Visual Basic for Tweens and Teens

Learn Computational and Algorithmic Thinking

Revised Second Edition

The Answers

By
Aristides S. Bouras

Visual Basic for Tweens and Teens – The Answers

Revised Second Edition

https://www.bouraspage.com

RCode: 220224

Warning and Disclaimer

This book is designed to provide information about learning "Computational and Algorithmic Thinking," mainly through the use of Visual Basic programming language. Every effort has been taken to make this book compatible with all releases of Visual Basic, and it is almost certain to be compatible with any future releases of Visual Basic.

The information is provided on an "as is" basis. The author shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the files that may accompany it.

Table of Contents

How to Report Errata

Although I have taken great care to ensure the accuracy of the content of this book, mistakes do occur. If you find a mistake in this book, either in the text or the code, I encourage you to report it to me. By doing so, you can save other readers from frustration and, of course, help me to improve the next release of this book. If you find any errata, please feel free to report them by visiting the following address:

https://www.bouraspage.com/report-errata

Once your errata are verified, your submission will be accepted and the errata will be uploaded to my website, and added to any existing list of errata.

Chapter 1

1.7 Review Questions: True/False

1. False
2. False
3. True
4. False
5. False
6. True
7. True
8. False
9. False
10. False
11. True
12. True
13. False
14. False
15. False
16. True
17. True
18. False
19. False
20. True
21. False
22. False
23. True

1.8 Review Questions: Multiple Choice

1. b
2. d
3. c
4. g
5. d
6. c
7. b
8. c
9. b
10. a

Chapter 3

3.11 Review Questions: True/False

1. True
2. False
3. False
4. False
5. True
6. False
7. True
8. True
9. True
10. False
11. True
12. False
13. True
14. False
15. False
16. False
17. True
18. False
19. False
20. False
21. False
22. True

3.12 Review Questions: Multiple Choice

1. a
2. c
3. c
4. a
5. a
6. d

Chapter 4

4.8 Review Questions: True/False

1. False
2. True
3. False
4. False
5. True
6. False
7. True
8. False
9. True
10. False
11. True
12. True
13. True
14. True
15. False

4.9 Review Questions: Multiple Choice

1. e
2. a
3. b
4. b
5. c
6. b
7. d
8. a

4.10 Review Exercises

1. 1 – c, 2 – d, 3 – a, 4 – b
2. 1 – d, 2 – c, 3 – b, 4 – a
3.

Value	Data Type	Declaration and Initialization
The name of my friend	String	`Dim name As String = "Mark"`
My address	String	`Dim address As String` `address = "254 Lookout Rd. Wilson, NY 27893"`
The average daily temperature	Float	`Dim average As Double = 70.3`
A telephone number	String	`Dim phone_number As String = "1-891-764-2410"`
My Social Security Number (SSN)	String	`Dim ssn As String = "123-45-6789"`
The speed of a car	Float	`Dim speed As Double = 90.5`
The number of children in a family	Integer	`Dim children As Integer = 3`

Chapter 5

5.4 Review Questions: True/False

1. True
2. True
3. True
4. False
5. False

5.5 Review Questions: Multiple Choice

1. a
2. c
3. b
4. b

Chapter 6

6.6 Review Questions: True/False

1. False
2. True
3. False
4. False
5. False
6. False
7. False
8. False
9. False
10. False
11. False
12. True
13. False
14. False
15. True
16. False
17. False
18. False
19. True
20. False

6.7 Review Questions: Multiple Choice

1. c
2. c
3. b
4. d
5. d
6. b
7. d
8. a

6.8 Review Exercises

1. ii, iv, v, ix, x
2. i – String, ii – Boolean, iii – String, iv – String, v – Float, vi – Integer
3. i – b, ii – d, iii – c, iv – e
4. i – 26, ii – 28
5. i – 0, ii – 4
6. i – 2.0, ii – 40
7. My name is Alexander the Great
8. i – 3, ii – 1
9. California California California

Chapter 8

8.2 Review Exercises

1. Solution

```
Sub Main(args As String())
    Dim b, h, area As Double

    Console.Write("Enter base: ")
    b = Console.ReadLine()
    Console.Write("Enter height: ")
    h = Console.ReadLine()

    area = b * h / 2

    Console.WriteLine(area)
```

```
End Sub
```

2. Solution

```
Sub Main(args As String())
    Dim f, k As Double

    Console.Write("Enter temperature in Fahrenheit: ")
    f = Console.ReadLine()

    k = (f + 459.67) / 1.8

    Console.WriteLine(k)
End Sub
```

3. Solution

```
Sub Main(args As String())
    Dim angle1, angle2, angle3 As Double

    Console.Write("Enter 1st angle: ")
    angle1 = Console.ReadLine()
    Console.Write("Enter 2nd angle: ")
    angle2 = Console.ReadLine()

    angle3 = 180 - angle1 - angle2

    Console.WriteLine(angle3)
End Sub
```

4. Solution

```
Sub Main(args As String())
    Dim average, g1, g2, g3, g4 As Double

    Console.Write("Enter 1st grade: ")
    g1 = Console.ReadLine()
    Console.Write("Enter 2nd grade: ")
    g2 = Console.ReadLine()
    Console.Write("Enter 3rd grade: ")
    g3 = Console.ReadLine()
    Console.Write("Enter 4th grade: ")
    g4 = Console.ReadLine()

    average = (g1 + g2 + g3 + g4) / 4.0

    Console.WriteLine(average)
End Sub
```

5. Solution

```
Const PI = 3.14159
```

```
Sub Main(args As String())
    Dim r, perimeter As Double

    Console.Write("Enter radius: ")
    r = Console.ReadLine()

    perimeter = 2 * PI * r

    Console.WriteLine(perimeter)
End Sub
```

6. Solution

```
Const PI = 3.14159

Sub Main(args As String())
    Dim d, radius, volume As Double

    Console.Write("Enter diameter (in meters): ")
    d = Console.ReadLine()

    radius = d / 2

    volume = 4 / 3 * PI * radius ^ 3

    Console.WriteLine(volume)
End Sub
```

7. Solution

Only a), e) and g) are syntactically correct. The latter is more user friendly.

8. Solution

```
Const PI = 3.14159

Sub Main(args As String())
    Dim d, radius, perimeter, area, volume As Double

    Console.Write("Enter diameter (in meters): ")
    d = Console.ReadLine()

    radius = d / 2
    perimeter = 2 * PI * radius
    area = PI * radius ^ 2

    volume = 4 / 3 * PI * radius ^ 3

    Console.WriteLine(radius & " " & perimeter & " " & area & " " & volume)
End Sub
```

9. Solution

```
Sub Main(args As String())
    Dim w, h As Integer
    Dim bmi As Double

    Console.Write("Enter weight in pounds: ")
    w = Console.ReadLine()
    Console.Write("Enter height in inches: ")
    h = Console.ReadLine()

    bmi = w * 703.0 / (h * h)

    Console.WriteLine(bmi)
End Sub
```

10.Solution

```
Sub Main(args As String())
    Dim d, m, days_passed, days_left As Integer

    Console.Write("Enter current month: ")
    m = Console.ReadLine()
    Console.Write("Enter current day: ")
    d = Console.ReadLine()

    days_passed = (m - 1) * 30 + d
    days_left = 360 - days_passed

    Console.WriteLine(days_left)
End Sub
```

11.Solution

```
Sub Main(args As String())
    Dim first_name, middle_name, last_name, title As String

    Console.Write("First name: ")
    first_name = Console.ReadLine()
    Console.Write("Middle name: ")
    middle_name = Console.ReadLine()
    Console.Write("Last name: ")
    last_name = Console.ReadLine()
    Console.Write("Title: ")
    title = Console.ReadLine()

    Console.WriteLine(title & " " & first_name & " " & middle_name & " " & last_name)
    Console.WriteLine(first_name & " " & middle_name & " " & last_name)
    Console.WriteLine(last_name & ", " & first_name)
    Console.WriteLine(last_name & ", " & first_name & " " & middle_name)
```

```
    Console.WriteLine(last_name & ", " & first_name & " " & middle_name & ", " & title)
    Console.WriteLine(first_name & " " & last_name)
End Sub
```

Chapter 9

9.3 Review Questions: True/False

1. True
2. False
3. False
4. True
5. False
6. True
7. False
8. False
9. True
10. True
11. False

9.4 Review Exercises

1. 2
2. i – 2.5, ii – 2.2
3. i – 4, ii – 9
4. i – 12, ii – 8.5
5. i – 5, ii – 4

6. Solution

```
Sub Main(args As String())
    Dim a, b, hypotenuse As Double

    Console.Write("Enter right angle side A of a right-angled triangle: ")
    a = Console.ReadLine()
    Console.Write("Enter right angle side B of a right-angled triangle: ")
    b = Console.ReadLine()

    hypotenuse = Math.Sqrt(a ^ 2 + b ^ 2)

    Console.WriteLine(hypotenuse)
End Sub
```

Chapter 10

10.4 Review Questions: True/False

1. True
2. False
3. False
4. True
5. True
6. True
7. False
8. False
9. True
10. False
11. True

10.5 Review Questions: Multiple Choice

1. d
2. b
3. a
4. b
5. c
6. a
7. c

10.6 Review Exercises

1. Solution

```
Sub Main(args As String())
    Dim alphabet As String
    Dim rnd As New Random()

    alphabet = "abcdefghijklmnopqrstuvwxyz"

    Console.Write(alphabet(rnd.Next(0, 26)).ToString().ToUpper())
    Console.Write(alphabet(rnd.Next(0, 26)))
    Console.Write(alphabet(rnd.Next(0, 26)))
    Console.Write(alphabet(rnd.Next(0, 26)))
End Sub
```

2. Solution

```
Sub Main(args As String())
    Dim name, x, secret_password As String
    Dim rnd As New Random()

    Console.Write("Enter name: ")
    name = Console.ReadLine()

    x = name.ToLower().Replace(" ", "")    'Convert to lower case and remove space

    secret_password = x(rnd.Next(0, x.Length)) & x(rnd.Next(0, x.Length)) &
                      x(rnd.Next(0, x.Length)) & rnd.Next(1000, 10000)

    Console.WriteLine(secret_password)
End Sub
```

3. Solution

First approach

```
Sub Main(args As String())
    Dim number, reversed_number As Integer
    Dim s_number, digit1, digit2, digit3 As String

    Console.Write("Enter a three-digit integer: ")
    number = Console.ReadLine()

    s_number = number   's_number is of type String

    digit1 = s_number(0)
```

```
    digit2 = s_number(1)
    digit3 = s_number(2)

    reversed_number = 100 * Convert.ToInt32(digit3) + 10 * Convert.ToInt32(digit2) +
                      Convert.ToInt32(digit1)

    Console.WriteLine(reversed_number)
End Sub
```

Second approach

```
Sub Main(args As String())
    Dim number, reversed_number As Integer
    Dim s_number As String

    Console.Write("Enter a three-digit integer: ")
    number = Console.ReadLine()

    s_number = number  's_number is of type String
    reversed_number = Convert.ToInt32(s_number(2) & s_number(1) & s_number(0))

    Console.WriteLine(reversed_number)
End Sub
```

Chapter 11

11.9 Review Questions: True/False

1. True
2. False
3. False
4. False
5. False
6. True
7. True
8. True
9. True
10. True
11. True
12. True
13. False
14. False
15. True
16. True
17. False
18. True
19. True

11.10 Review Questions: Multiple Choice

1. b
2. a
3. a
4. a
5. c

11.11 Review Exercises

1. Solution

i. b, d, f

ii. i

iii. c, e

iv. a, g, h

2. Solution

a	b	c	a <> 1	b > a	c / 2 > 2 * a
3	–5	8	*True*	*False*	*False*
1	10	20	*False*	*True*	*True*
–4	–2	–9	*True*	*True*	*True*

3. Solution

BE1 (Boolean Expression 1)	*BE2* (Boolean Expression 2)	*BE1 Or BE2*	*BE1 And BE2*	Not(BE2)
False	False	*False*	*False*	*True*
False	True	*True*	*False*	*False*
True	False	*True*	*False*	*True*
True	True	*True*	*True*	*False*

4. Solution

a	b	c	a > 3 Or c > b And c > 1	a > 3 And c > b Or c > 1
4	–6	2	*True*	*True*
–3	2	–4	*False*	*False*

5. Solution

Expression	Value
`(x + y) ^ 3`	*8.0*
`(x + y) / (x ^ 2 – 14)`	*1.0*
`(x – 1) = y + 5`	*True*
`x > 2 And y = 1`	*False*
`x = 1 Or Not(flag = False)`	*True*

6. Solution

a. `age < 12 And age <> 8`

b. `age >= 6 And age <= 9 Or age = 11`

c. `age > 7 And age <> 10 And age <> 12`

d. `age = 6 Or age = 9 Or age = 11`

e. `age >= 6 And age <= 12 And age <> 8`

f. `age <> 7 And age <> 10`

Chapter 12

12.2 Review Questions: True/False

1. False
2. False
3. True
4. False

12.3 Review Questions: Multiple Choice

1. b
2. a
3. d
4. c

12.4 Review Exercises

1. Solution

```
Sub Main(args As String())
    Dim x, y As Double

    x = Console.ReadLine()

    y = -5
    If x * y / 2 > 20
        y *= 2
        x += 4 * x ^ 2
    End If
    Console.WriteLine(x & " " & y)
End Sub
```

2. Solution

i. 9 12

ii. 2 2

3. Solution

```
Sub Main(args As String())
    Dim x As Double

    Console.Write("Enter a number: ")
    x = Console.ReadLine()

    If x > 0 Then
        Console.WriteLine("Positive")
    End If
End Sub
```

4. Solution

```
Sub Main(args As String())
    Dim x, y As Double

    Console.Write("Enter a number: ")
    x = Console.ReadLine()
```

```
    Console.Write("Enter a second number")
    y = Console.ReadLine()

    If x > 0 And y > 0 Then
        Console.WriteLine("Positives")
    End If
End Sub
```

5. Solution

```
Sub Main(args As String())
    Dim x As Integer

    Console.Write("Enter your age: ")
    x = Console.ReadLine()

    If x > 14 Then
        Console.WriteLine("You can drive a car in Kansas (USA)")
    End If
End Sub
```

6. Solution

```
Sub Main(args As String())
    Dim s As String

    Console.Write("Enter a string: ")
    s = Console.ReadLine()

    If s = s.ToUpper() Then
        Console.WriteLine("Uppercase")
    End If
End Sub
```

7. Solution

```
Sub Main(args As String())
    Dim s As String

    Console.Write("Enter a string: ")
    s = Console.ReadLine()

    If s.Length > 20 Then
        Console.WriteLine("Many characters")
    End If
End Sub
```

8. Solution

```
Sub Main(args As String())
    Dim n1, n2, n3 As Double

    Console.Write("Enter 1st number: ")
```

```
    n1 = Console.ReadLine()
    Console.Write("Enter 2nd number: ")
    n2 = Console.ReadLine()
    Console.Write("Enter 3rd number: ")
    n3 = Console.ReadLine()

    If n1 < 0 Or n2 < 0 Or n3 < 0 Then
        Console.WriteLine("Among the given numbers, there is a negative one!")
    End If
End Sub
```

9. Solution

```
Sub Main(args As String())
    Dim t1, t2, t3, average As Double

    Console.Write("Enter 1st temperature: ")
    t1 = Console.ReadLine()
    Console.Write("Enter 2nd temperature: ")
    t2 = Console.ReadLine()
    Console.Write("Enter 3rd temperature: ")
    t3 = Console.ReadLine()

    average = (t1 + t2 + t3) / 3

    If average > 60 Then
        Console.WriteLine("Heat Wave")
    End If
End Sub
```

10.Solution

```
Sub Main(args As String())
    Dim w1, w2, w3, w4, maximum As Double

    Console.Write("Enter the weight of the 1st person: ")
    w1 = Console.ReadLine()
    Console.Write("Enter the weight of the 2nd person: ")
    w2 = Console.ReadLine()
    Console.Write("Enter the weight of the 3rd person: ")
    w3 = Console.ReadLine()
    Console.Write("Enter the weight of the 4th person: ")
    w4 = Console.ReadLine()

    maximum = w1

    If w2 > maximum Then
        maximum = w2
    End If
```

```
    If w3 > maximum Then
        maximum = w3
    End If

    If w4 > maximum Then
        maximum = w4
    End If

    Console.WriteLine(maximum)
End Sub
```

11.Solution

```
Sub Main(args As String())
    Dim n1, n2, n3, n4, m_name As String
    Dim a1, a2, a3, a4, minimum As Integer

    Console.Write("Enter the age of the 1st person: ")
    a1 = Console.ReadLine()
    Console.Write("Enter the name of the 1st person: ")
    n1 = Console.ReadLine()

    Console.Write("Enter the age of the 2nd person: ")
    a2 = Console.ReadLine()
    Console.Write("Enter the name of the 2nd person: ")
    n2 = Console.ReadLine()

    Console.Write("Enter the age of the 3rd person: ")
    a3 = Console.ReadLine()
    Console.Write("Enter the name of the 3rd person: ")
    n3 = Console.ReadLine()

    Console.Write("Enter the age of the 4th person: ")
    a4 = Console.ReadLine()
    Console.Write("Enter the name of the 4th person: ")
    n4 = Console.ReadLine()

    minimum = a1
    m_name = n1

    If a2 < minimum Then
        minimum = a2
        m_name = n2
    End If

    If a3 < minimum Then
        minimum = a3
        m_name = n3
    End If
```

```
    If a4 < minimum Then
        minimum = a4
        m_name = n4
    End If

    Console.WriteLine("The youngest person is " & m_name)
End Sub
```

12.Solution

```
Sub Main(args As String())
    Dim a1, a2, a3, minimum, maximum, middle As Integer

    Console.Write("Enter the age of the 1st person: ")
    a1 = Console.ReadLine()
    Console.Write("Enter the age of the 2nd person: ")
    a2 = Console.ReadLine()
    Console.Write("Enter the age of the 3rd person: ")
    a3 = Console.ReadLine()

    minimum = a1
    If a2 < minimum Then
        minimum = a2
    End If

    If a3 < minimum Then
        minimum = a3
    End If

    maximum = a1
    If a2 > maximum Then
        maximum = a2
    End If

    If a3 > maximum Then
        maximum = a3
    End If

    middle = a1 + a2 + a3 - minimum - maximum

    Console.WriteLine(middle)
End Sub
```

Chapter 13

13.2 Review Questions: True/False

1. False
2. True
3. False
4. False

13.3 Review Questions: Multiple Choice

1. a
2. a
3. a
4. d
5. c

13.4 Review Exercises

1. Solution

i. 1
ii. 5

2. Solution

i. 7.0 18.0
ii. 0.5 3.5

3. Solution

```
Sub Main(args As String())
    Dim num As Double

    Console.Write("Enter a number: ")
    num = Console.ReadLine()

    If num > 100 Then
        Console.WriteLine("Given number is greater than 100")
    Else
        Console.WriteLine("Given number is less than or equal to 100")
    End If
End Sub
```

4. Solution

```
Sub Main(args As String())
    Dim num As Double

    Console.Write("Enter a number: ")
    num = Console.ReadLine()

    If num >= 0 And num <= 100 Then
        Console.WriteLine("Given number is between 0 and 100")
    Else
        Console.WriteLine("Given number is not between 0 and 100")
    End If
End Sub
```

5. Solution

```
Sub Main(args As String())
    Dim num As Integer

    num = Console.ReadLine()
```

```
    If num >= 1000 And num <= 9999 Then
        Console.WriteLine(num & " is a four-digit integer")
    Else
        Console.WriteLine(num & " is not a four-digit integer")
    End If

End Sub
```

6. Solution

```
Sub Main(args As String())
    Dim num1, num2 As Double

    Console.Write("Enter first number: ")
    num1 = Console.ReadLine()
    Console.Write("Enter second number: ")
    num2 = Console.ReadLine()

    If num1 < num2 Then
        Console.WriteLine(num1)
    Else
        Console.WriteLine(num2)
    End If
End Sub
```

7. Solution

```
Sub Main(args As String())
    Dim name1, name2 As String
    Dim goals1, goals2 As Integer

    Console.Write("Enter team name 1: ")
    name1 = Console.ReadLine()
    Console.Write("Enter team name 2: ")
    name2 = Console.ReadLine()

    Console.Write("Enter goals " & name1 & " scored: ")
    goals1 = Int32.Parse(Console.ReadLine())
    Console.Write("Enter goals " & name2 & " scored: ")
    goals2 = Int32.Parse(Console.ReadLine())

    If goals1 > goals2 Then
        Console.WriteLine("Winner: " & name1)
    Else
        Console.WriteLine("Winner: " & name2)
    End If
End Sub
```

8. Solution

```
Sub Main(args As String())
    Dim a, b, c, average As Double

    Console.Write("Enter 1st jump in meters: ")
    a = Console.ReadLine()
    Console.Write("Enter 2nd jump in meters: ")
    b = Console.ReadLine()
    Console.Write("Enter 3rd jump in meters: ")
    c = Console.ReadLine()

    average = (a + b + c) / 3

    If average >= 8 Then
        Console.WriteLine("Qualified")
    Else
        Console.WriteLine("Disqualified")
    End If
End Sub
```

Chapter 14

14.2 Review Questions: True/False

1. True
2. False
3. False
4. False
5. False
6. True
7. True

14.3 Review Exercises

1. Solution

i. 1 ii. 2 iii. 4 iv. 4

2. Solution

i. 0 5.0 ii. 10 90.0 iii. 20 160.0

3. Solution

```
Sub Main(args As String())
    Dim name1, name2 As String
    Dim goals1, goals2 As Integer

    Console.Write("Enter team name 1: ")
    name1 = Console.ReadLine()
    Console.Write("Enter team name 2: ")
    name2 = Console.ReadLine()

    Console.Write("Enter goals " & name1 & " scored: ")
```

```
    goals1 = Int32.Parse(Console.ReadLine())
    Console.Write("Enter goals " & name2 & " scored: ")
    goals2 = Int32.Parse(Console.ReadLine())

    If goals1 > goals2 Then
        Console.WriteLine("Winner: " & name1)
    ElseIf goals2 > goals1 Then
        Console.WriteLine("Winner: " & name2)
    Else
        Console.WriteLine("It's a tie!")
    End If
End Sub
```

4. Solution

First approach

```
Sub Main(args As String())
    Dim a, n As Integer

    Console.Write("Enter an integer between -9999 and 9999: ")
    a = Console.ReadLine()

    If a >= -9999 And a <= -1000 Or a >= 1000 And a <= 9999 Then
        n = 4
    ElseIf a >= -999 And a <= -100 Or a >= 100 And a <= 999 Then
        n = 3
    ElseIf a >= -99 And a <= -10 Or a >= 10 And a <= 99 Then
        n = 2
    Else
        n = 1
    End If

    Console.Write("You entered a " & n & "-digit integer")
End Sub
```

Second approach

```
Sub Main(args As String())
    Dim a, n As Integer

    Console.Write("Enter an integer between -9999 and 9999: ")
    a = Console.ReadLine()

    'If variable a is negative, make it positive
    If a < 0 Then
        a = (-1) * a
    End If

    If a >= 1000 And a <= 9999 Then
        n = 4
```

```
    ElseIf a >= 100 And a <= 999 Then
        n = 3
    ElseIf a >= 10 And a <= 99 Then
        n = 2
    Else
        n = 1
    End If

    Console.Write("You entered a " & n & "-digit integer")
End Sub
```

5. Solution

First approach

```
Sub Main(args As String())
    Dim a As Integer

    Console.Write("Enter an integer between -9999 and 9999: ")
    a = Console.ReadLine()

    If a >= -9999 And a <= -1000 Or a >= 1000 And a <= 9999 Then
        Console.WriteLine("You entered a 4-digit integer")
    ElseIf a >= -999 And a <= -100 Or a >= 100 And a <= 999 Then
        Console.WriteLine("You entered a 3-digit integer")
    ElseIf a >= -99 And a <= -10 Or a >= 10 And a <= 99 Then
        Console.WriteLine("You entered a 2-digit integer")
    ElseIf a >= -9 And a <= -1 Or a >= 1 And a <= 9 Then
        Console.WriteLine("You entered a 1-digit integer")
    Else
        Console.WriteLine("Error: Invalid value!")
    End If
End Sub
```

Second approach

```
Sub Main(args As String())
    Dim a As Integer

    Console.Write("Enter an integer between -9999 and 9999: ")
    a = Console.ReadLine()

    'If variable a is negative, make it positive
    If a < 0 Then
        a = (-1) * a
    End If

    If a >= 1000 And a <= 9999 Then
        Console.WriteLine("You entered a 4-digit integer")
    ElseIf a >= 100 And a <= 999 Then
        Console.WriteLine("You entered a 3-digit integer")
```

```
    ElseIf a >= 10 And a <= 99 Then
        Console.WriteLine("You entered a 2-digit integer")
    ElseIf a >= 1 And a <= 9 Then
        Console.WriteLine("You entered a 1-digit integer")
    Else
        Console.WriteLine("Error: Invalid value!")
    End If
End Sub
```

6. Solution

```
Sub Main(args As String())
    Dim m As Integer

    Console.Write("Enter the number of a month between 1 and 12: ")
    m = Console.ReadLine()

    If m <= 2 Or m = 12 Then
        Console.WriteLine("Winter")
    ElseIf m <= 5 Then
        Console.WriteLine("Spring")
    ElseIf m <= 8 Then
        Console.WriteLine("Summer")
    Else
        Console.WriteLine("Fall (Autumn)")
    End If
End Sub
```

7. Solution

```
Sub Main(args As String())
    Dim m As Integer

    Console.Write("Enter the number of a month between 1 and 12: ")
    m = Console.ReadLine()

    If m < 1 Or m > 12 Then
        Console.WriteLine("Error: Invalid value!")
    ElseIf m <= 2 Or m = 12 Then
        Console.WriteLine("Winter")
    ElseIf m <= 5 Then
        Console.WriteLine("Spring")
    ElseIf m <= 8 Then
        Console.WriteLine("Summer")
    Else
        Console.WriteLine("Fall (Autumn)")
    End If
End Sub
```

8. Solution

```
Sub Main(args As String())
    Dim name As String

    Console.Write("Enter the name of a month: ")
    name = Console.ReadLine().ToUpper()

    If name = "JANUARY" Then
        Console.WriteLine(1)
    ElseIf name = "FEBRUARY" Then
        Console.WriteLine(2)
    ElseIf name = "MARCH" Then
        Console.WriteLine(3)
    ElseIf name = "APRIL" Then
        Console.WriteLine(4)
    ElseIf name = "MAY" Then
        Console.WriteLine(5)
    ElseIf name = "JUNE" Then
        Console.WriteLine(6)
    ElseIf name = "JULY" Then
        Console.WriteLine(7)
    ElseIf name = "AUGUST" Then
        Console.WriteLine(8)
    ElseIf name = "SEPTEMBER" Then
        Console.WriteLine(9)
    ElseIf name = "OCTOBER" Then
        Console.WriteLine(10)
    ElseIf name = "NOVEMBER" Then
        Console.WriteLine(11)
    ElseIf name = "DECEMBER" Then
        Console.WriteLine(12)
    Else
        Console.WriteLine("Error")
    End If
End Sub
```

9. Solution

```
Sub Main(args As String())
    Dim letter As String

    Console.Write("Enter a letter between A and F: ")
    letter = Console.ReadLine()

    If letter = "A" Then
        Console.WriteLine("90 - 100")
    ElseIf letter = "B" Then
```

```
        Console.WriteLine("80 - 89")
    ElseIf letter = "C" Then
        Console.WriteLine("70 - 79")
    ElseIf letter = "D" Then
        Console.WriteLine("60 - 69")
    Else
        Console.WriteLine("0 - 59")
    End If
End Sub
```

10.Solution

```
Sub Main(args As String())
    Dim roman As String

    Console.Write("Enter a Roman numeral between I and X: ")
    roman = Console.ReadLine()

    If roman = "I" Then
        Console.WriteLine(1)
    ElseIf roman = "II" Then
        Console.WriteLine(2)
    ElseIf roman = "III" Then
        Console.WriteLine(3)
    ElseIf roman = "IV" Then
        Console.WriteLine(4)
    ElseIf roman = "V" Then
        Console.WriteLine(5)
    ElseIf roman = "VI" Then
        Console.WriteLine(6)
    ElseIf roman = "VII" Then
        Console.WriteLine(7)
    ElseIf roman = "VIII" Then
        Console.WriteLine(8)
    ElseIf roman = "IX" Then
        Console.WriteLine(9)
    ElseIf roman = "X" Then
        Console.WriteLine(10)
    Else
        Console.WriteLine("Error")
    End If
End Sub
```

11.Solution

```
Sub Main(args As String())
    Dim total As Integer

    Console.Write("Enter the total number of CDs purchased in a month: ")
```

```
    total = Console.ReadLine()

    If total = 1 Then
        Console.WriteLine("You are awarded 3 points")
    ElseIf total = 2 Then
        Console.WriteLine("You are awarded 10 points")
    ElseIf total = 3 Then
        Console.WriteLine("You are awarded 20 points")
    Else
        Console.WriteLine("You are awarded 45 points")
    End If
End Sub
```

12.Solution

```
Sub Main(args As String())
    Dim num As String

    Console.Write("Enter a number (0 - 3) in words")
    num = Console.ReadLine()

    If num = "zero" Then
        Console.WriteLine(0)
    ElseIf num = "one" Then
        Console.WriteLine(1)
    ElseIf num = "two" Then
        Console.WriteLine(2)
    ElseIf num = "three" Then
        Console.WriteLine(3)
    Else
        Console.WriteLine("I don't know this number!")
    End If
End Sub
```

13.Solution

```
Sub Main(args As String())
    Dim b As Integer

    Console.Write("Enter Beaufort number: ")
    b = Console.ReadLine()

    If b = 0 Then
        Console.WriteLine("Calm")
    ElseIf b = 1 Then
        Console.WriteLine("Light Air")
    ElseIf b = 2 Then
        Console.WriteLine("Light breeze")
    ElseIf b = 3 Then
```

```
        Console.WriteLine("Gentle breeze")
    ElseIf b = 4 Then
        Console.WriteLine("Moderate breeze")
    ElseIf b = 5 Then
        Console.WriteLine("Fresh breeze")
    ElseIf b = 6 Then
        Console.WriteLine("Strong breeze")
    ElseIf b = 7 Then
        Console.WriteLine("Moderate gale")
    ElseIf b = 8 Then
        Console.WriteLine("Gale")
    ElseIf b = 9 Then
        Console.WriteLine("Strong gale")
    ElseIf b = 10 Then
        Console.WriteLine("Storm")
    ElseIf b = 11 Then
        Console.WriteLine("Violent storm")
    ElseIf b = 12 Then
        Console.WriteLine("Hurricane force")
    Else
        Console.WriteLine("Invalid Beaufort number!")
    End If
End Sub
```

14.Solution

```
Sub Main(args As String())
    Dim wind As Double

    Console.Write("Enter wind speed (in miles/hour): ")
    wind = Console.ReadLine()

    If wind < 0 Then
        Console.WriteLine("Entered value is negative")
    ElseIf wind < 1 Then
        Console.Write("Beaufort: 0" & vbCrLf & "Calm")
    ElseIf wind < 4 Then
        Console.Write("Beaufort: 1" & vbCrLf & "Light air")
    ElseIf wind < 8 Then
        Console.Write("Beaufort: 2" & vbCrLf & "Light breeze")
    ElseIf wind < 13 Then
        Console.Write("Beaufort: 3" & vbCrLf & "Gentle breeze")
    ElseIf wind < 18 Then
        Console.Write("Beaufort: 4" & vbCrLf & "Moderate breeze")
    ElseIf wind < 25 Then
        Console.Write("Beaufort: 5" & vbCrLf & "Fresh breeze")
    ElseIf wind < 31 Then
```

```
        Console.Write("Beaufort: 6" & vbCrLf & "Strong breeze")
    ElseIf wind < 39 Then
        Console.Write("Beaufort: 7" & vbCrLf & "Moderate gale")
    ElseIf wind < 47 Then
        Console.Write("Beaufort: 8" & vbCrLf & "Gale")
    ElseIf wind < 55 Then
        Console.Write("Beaufort: 9" & vbCrLf & "Strong gale")
    ElseIf wind < 64 Then
        Console.Write("Beaufort: 10" & vbCrLf & "Storm")
    ElseIf wind < 74 Then
        Console.Write("Beaufort: 11" & vbCrLf & "Violent storm")
    Else
        Console.Write("Beaufort: 12" & vbCrLf & "Hurricane force")
    End If
End Sub
```

15.Solution

```
Sub Main(args As String())
    Dim choice As Integer
    Dim kelvin, fahrenheit, celsius As Double

    Console.WriteLine("1. Convert Kelvin to Fahrenheit")
    Console.WriteLine("2. Convert Fahrenheit to Kelvin")
    Console.WriteLine("3. Convert Fahrenheit to Celsius")
    Console.WriteLine("4. Convert Celsius to Fahrenheit")

    Console.Write("Enter a choice: ")
    choice = Console.ReadLine()

    If choice = 1 Then
        Console.Write("Enter a temperature in degrees Kelvin: ")
        kelvin = Console.ReadLine()
        fahrenheit = 1.8 * kelvin - 459.67
        Console.Write(kelvin & " degrees Kelvin = " & fahrenheit & " degrees Fahrenheit")
    ElseIf choice = 2 Then
        Console.Write("Enter a temperature in degrees Fahrenheit: ")
        fahrenheit = Console.ReadLine()
        kelvin = (fahrenheit + 459.67) / 1.8
        Console.Write(fahrenheit & " degrees Fahrenheit = " & kelvin & " degrees Kelvin")
    ElseIf choice = 3 Then
        Console.Write("Enter a temperature in degrees Fahrenheit: ")
        fahrenheit = Console.ReadLine()
        celsius = 5 / 9 * (fahrenheit - 32)
        Console.Write(fahrenheit & " degrees Fahrenheit = " & celsius & " degrees
Celsius")
    ElseIf choice = 4 Then
```

```
        Console.Write("Enter a temperature in degrees Celsius: ")
        celsius = Console.ReadLine()
        fahrenheit = 9 / 5 * celsius + 32
        Console.Write(celsius & " degrees Celsius = " & " degrees Fahrenheit")
    Else
        Console.WriteLine("Invalid choice!")
    End If
End Sub
```

Chapter 15

15.2 Review Questions: True/False

1. True
2. True
3. False

15.3 Review Exercises

1. Solution

i. 25 6 ii. 10 9 iii. 50 2

2. Solution

First approach

```
Sub Main(args As String())
    Dim age As Integer

    Console.Write("Enter your age: ")
    age = Console.ReadLine()

    If age < 0 Then
        Console.WriteLine("Error: Invalid age!")
    Else
        If age < 16 Then
            Console.WriteLine("You cannot drive either a small scooter or a car")
        Else
            If age < 18 Then
                Console.WriteLine("You can drive a small scooter")
            Else
                Console.WriteLine("You can drive a car and a small scooter")
            End If
        End If
    End If
End Sub
```

Second approach

```
Sub Main(args As String())
    Dim age As Integer
```

```
    Console.Write("Enter your age: ")
    age = Console.ReadLine()

    If age < 0 Then
        Console.WriteLine("Error: Invalid age!")
    Else
        If age < 16 Then
            Console.WriteLine("You cannot drive either a small scooter or a car")
        ElseIf age < 18 Then
            Console.WriteLine("You can drive a small scooter")
        Else
            Console.WriteLine("You can drive a car and a small scooter")
        End If
    End If
End Sub
```

Third approach

```
Sub Main(args As String())
    Dim age As Integer

    Console.Write("Enter your age: ")
    age = Console.ReadLine()

    If age < 0 Then
        Console.WriteLine("Error: Invalid age!")
    ElseIf age < 16 Then
        Console.WriteLine("You cannot drive either a small scooter or a car")
    ElseIf age < 18 Then
        Console.WriteLine("You can drive a small scooter")
    Else
        Console.WriteLine("You can drive a car and a small scooter")
    End If
End Sub
```

3. Solution

First approach

```
Sub Main(args As String())
    Dim t, w As Double

    Console.Write("Enter temperature (in Fahrenheit): ")
    t = Console.ReadLine()
    Console.Write("Enter wind speed (in miles/hour): ")
    w = Console.ReadLine()

    If t > 75 Then
        If w > 12 Then
            Console.WriteLine("The day is hot and windy")
        Else
```

```
            Console.WriteLine("The day is hot and not windy")
        End If
    Else
        If w > 12 Then
            Console.WriteLine("The day is cold and windy")
        Else
            Console.WriteLine("The day is cold and not windy")
        End If
    End If
End Sub
```

Second approach

```
Sub Main(args As String())
    Dim t, w As Double
    Dim message1, message2 As String

    Console.Write("Enter temperature (in Fahrenheit): ")
    t = Console.ReadLine()
    Console.Write("Enter wind speed (in miles/hour): ")
    w = Console.ReadLine()

    If t > 75 Then
        message1 = "hot"
    Else
        message1 = "cold"
    End If

    If w > 12 Then
        message2 = "windy"
    Else
        message2 = "not windy"
    End If

    Console.WriteLine("The day is " & message1 & " and " & message2)
End Sub
```

4. Solution

```
Sub Main(args As String())
    Dim a, w, h As Integer
    Dim bmi As Double

    Console.Write("Enter age: ")
    a = Console.ReadLine()
    If a < 18 Then
        Console.WriteLine("Invalid age")
    Else
        Console.Write("Enter weight in pounds: ")
        w = Console.ReadLine()
```

```
        Console.Write("Enter height in inches: ")
        h = Console.ReadLine()

        bmi = w * 703 / h ^ 2

        If bmi < 15 Then
            Console.WriteLine("Very severely underweight")
        ElseIf bmi < 16 Then
            Console.WriteLine("Severely underweight")
        ElseIf bmi < 18.5 Then
            Console.WriteLine("Underweight")
        ElseIf bmi < 25 Then
            Console.WriteLine("Normal")
        ElseIf bmi < 30 Then
            Console.WriteLine("Overweight")
        ElseIf bmi < 35 Then
            Console.WriteLine("Severely overweight")
        Else
            Console.WriteLine("Very severely overweight")
        End If
    End If
End Sub
```

Chapter 16

16.3 Review Questions: True/False

1. True
2. True
3. False
4. True

Chapter 17

17.3 Review Questions: True/False

1. True
2. False
3. False
4. False
5. False
6. False
7. False
8. False
9. True
10. True
11. True

17.4 Review Questions: Multiple Choice

1. b
2. b
3. c
4. b
5. c
6. b
7. c
8. a
9. b
10. c
11. b

17.5 Review Exercises

1. Solution

```
Dim i As Double = 30
Do While i > 5
    Console.WriteLine(i)
    i /= 2
Loop
Console.WriteLine("The end")
```

2. Solution

```
Dim i = 3 As Integer = 3


Do
    i -= 1
Loop While i > 0
Console.Write("The end")
```

3. Solution

Four

4. Solution

Zero

5. Solution

It displays

2

14

6

and performs three iterations

6. Solution

i. –1
ii. 9
iii. 0.5
iv. –7
v. A value between 17 and 32
vi. 1.4

7. Solution

i. –1
ii. 18
iii. 0.5

iv. –20

v. 128

vi. 11.25

8. Solution

i. 4

ii. –2

iii. 2

iv. 10

9. Solution

```
Sub Main(args As String())
    Dim x, total As Double
    Dim i As Integer

    total = 0

    i = 1
    Do While i <= 20
        Console.Write("Enter a number: ")
        x = Console.ReadLine()
        total += x
        i += 1
    Loop
    Console.WriteLine(total / 20)
End Sub
```

10.Solution

```
Sub Main(args As String())
    Dim n, i As Integer
    Dim p, x As Double

    Console.Write("Enter N: ")
    n = Console.ReadLine()

    p = 1
    i = 1
    Do While i <= n
        Console.Write("Enter a number: ")
        x = Console.ReadLine()
        If x > 0 Then
            p *= x
        End If
        i += 1
    Loop
    Console.WriteLine(p)
```

```
End Sub
```

11.Solution

```
Sub Main(args As String())
    Dim i, x, total As Integer

    total = 0

    i = 1
    Do While i <= 10
        Console.Write("Enter an integer: ")
        x = Console.ReadLine()
        If x >= 100 And x <= 200 Then
            total += x
        End If
        i += 1
    Loop
    Console.WriteLine(total)
End Sub
```

12.Solution

```
Sub Main(args As String())
    Dim i, x, total As Integer

    total = 0
    i = 1
    Do While i <= 20
        Console.Write("Enter an integer: ")
        x = Console.ReadLine()
        If x >= 100 And x <= 999 Then
            total += x
        End If
        i += 1
    Loop
    Console.WriteLine(total)
End Sub
```

13.Solution

```
Sub Main(args As String())
    Dim x, p As Double

    p = 1
    Console.Write("Enter a number: ")
    x = Console.ReadLine()
    Do While x <> 0
        p *= x
        Console.Write("Enter a number: ")
```

```
        x = Console.ReadLine()
    Loop
    Console.WriteLine(p)
End Sub
```

Chapter 18

18.3 Review Questions: True/False

1. True
2. True
3. False
4. False
5. True
6. True
7. False
8. False
9. True

18.4 Review Questions: Multiple Choice

1. d
2. a
3. b
4. a
5. b
6. d
7. c
8. c
9. a

18.5 Review Exercises

1. Solution

It displays

12 3

and performs five iterations

2. Solution

It displays

10 4

19 20

28 32

3. Solution

i. 9

ii. Any value greater than or equal to 2 and less than 2.5

iii. −7 (or −6)

iv. −1

4. Solution

```
Sub Main(args As String())
    Dim i As Integer
    Dim x, p, total As Double

    p = 1
```

```
    total = 0
    For i = 1 To 20
        Console.Write("Enter a number: ")
        x = Console.ReadLine()
        p *= x
        total += x
    Next
    Console.WriteLine(p & " " & total / 20)
End Sub
```

5. Solution

```
Sub Main(args As String())
    Dim n, i, count, x As Integer

    Console.Write("Enter N: ")
    n = Console.ReadLine()

    count = 0
    For i = 1 To n
        Console.Write("Enter an integer: ")
        x = Console.ReadLine()
        If x > 0 Then
            count += 1
        End If
    Next

    If count > 0 Then
        Console.WriteLine(count)
    Else
        Console.WriteLine("You entered no positive integers")
    End If
End Sub
```

6. Solution

```
Sub Main(args As String())
    Dim i, start, finish As Integer

    Console.Write("Enter value for start: ")
    start = Console.ReadLine()
    Console.Write("Enter value for finish: ")
    finish = Console.ReadLine()

    For i = start To finish
        Console.WriteLine(i)
    Next
End Sub
```

7. Solution

```
Sub Main(args As String())
    Dim exp, i As Integer
    Dim p, b As Double

    Console.Write("Enter a value for base: ")
    b = Console.ReadLine()
    Console.Write("Enter an integer for exponent: ")
    exp = Console.ReadLine()

    p = 1
    For i = 1 To exp
        p *= b
    Next

    Console.WriteLine(p)
End Sub
```

8. Solution

First approach

```
Sub Main(args As String())
    Dim msg As String
    Dim count, words As Integer

    Console.Write("Enter a message: ")
    msg = Console.ReadLine()

    count = 0
    For Each ch In msg
        If ch = " " Then
            count += 1
        End If
    Next
    words = count + 1

    Console.WriteLine("The message entered contains " & words & " words")
End Sub
```

Second approach

```
Sub Main(args As String())
    Dim msg As String
    Dim i, count, words As Integer

    Console.Write("Enter a message: ")
    msg = Console.ReadLine()

    count = 0
    For i = 0 To msg.Length - 1
```

```
        If msg(i) = " " Then
            count += 1
        End If
    Next
    words = count + 1

    Console.WriteLine("The message entered contains " & words & " words")
End Sub
```

Chapter 19

19.2 Review Questions: True/False

1. True
2. False
3. True
4. False
5. True
6. True

19.3 Review Questions: Multiple Choice

1. b
2. c
3. d
4. a
5. b

19.4 Review Exercises

1. Solution

i. 10

ii. A value greater than or equal to 4.5 and less than 5.0

iii. −7 (or −8)

iv. 138 (or 139)

2. Solution

```
Sub Main(args As String())
    Dim hour, minutes As Integer

    For hour = 0 To 23
        For minutes = 0 To 59
            Console.WriteLine(hour & vbTab & minutes)
        Next
    Next
End Sub
```

3. Solution

```
Sub Main(args As String())
    Dim i, j As Integer

    For i = 5 To 1 Step -1
        For j = 1 To i
```

```
            Console.Write(i & " ")
        Next
        Console.WriteLine()
    Next
End Sub
```

4. Solution

```
Sub Main(args As String())
    Dim i, j As Integer

    For i = 0 To 5
        For j = 0 To i
            Console.Write(j & " ")
        Next
        Console.WriteLine()
    Next
End Sub
```

5. Solution

First approach (The amateur way!!!)

```
Sub Main(args As String())
    Console.WriteLine("*  *  *  *  *  *  *  *  *  *")
    Console.WriteLine("*  *  *  *  *  *  *  *  *  *")
    Console.WriteLine("*  *  *  *  *  *  *  *  *  *")
    Console.WriteLine("*  *  *  *  *  *  *  *  *  *")
End Sub
```

Second approach

```
Sub Main(args As String())
    Dim i, j As Integer

    For i = 1 To 4
        For j = 1 To 10
            Console.Write("*  ")
        Next
        Console.WriteLine()
    Next
End Sub
```

6. Solution

```
Sub Main(args As String())
    Dim y, i, j As Integer

    Console.Write("Enter an integer between 3 and 20: ")
    y = Console.ReadLine()

    For i = 1 To y
        For j = 1 To y
```

```
            Console.Write("*  ")
        Next
        Console.WriteLine()
    Next
End Sub
```

7. Solution

```
Sub Main(args As String())
    Dim i, j As Integer

    For i = 1 To 5
        For j = 1 To i
            Console.Write("*  ")
        Next
        Console.WriteLine()
    Next
End Sub
```

8. Solution

```
Sub Main(args As String())
    Dim i, j As Integer

    For i = 1 To 5
        For j = 1 To i
            Console.Write("*  ")
        Next
        Console.WriteLine()
    Next

    For i = 4 To 1 Step -1
        For j = 1 To i
            Console.Write("*  ")
        Next
        Console.WriteLine()
    Next
End Sub
```

Chapter 20

20.7 Review Questions: True/False

1. False
2. False
3. False
4. False
5. True
6. True
7. False
8. False

20.8 Review Questions: Multiple Choice

1. b
2. c
3. b
4. a
5. a
6. d

20.9 Review Exercises

1. Solution

```
count_names = 0
count_not_johns = 0
name = ""
Console.Write("Enter a name: ")
name = Console.ReadLine()
Do While name <> "STOP"
    Console.Write("Enter a name: ")
    name = Console.ReadLine()
    count_names += 1
    If name <> "John" Then
        count_not_johns += 1
    End If
    Console.Write("Enter a name: ")
    name = Console.ReadLine()
Loop
Console.WriteLine("Total names entered: " & count_names)
Console.WriteLine("Names other than John entered: " & count_not_johns)
```

2. Solution

```
Sub Main(args As String())
    Dim text As String
    Dim found As Boolean

    Console.Write("Enter a text: ")
    text = Console.ReadLine()

    found = False
    For Each character In text
        If character = " " Then
            found = True
            Exit For
        End If
    Next

    If found = False Then
        Console.WriteLine("One Single Word")
    Else
        Console.WriteLine("Complete Sentence")
    End If
End Sub
```

3. Solution

First approach

```
Sub Main(args As String())
    Dim sentence As String

    Dim found As Boolean
    Console.Write("Enter a sentence: ")
    sentence = Console.ReadLine()

    found = False
    For Each character In sentence
        If "0123456789".IndexOf(character) > -1 Then
            found = True
            Exit For
        End If
    Next

    If found = True Then
        Console.WriteLine("The sentence contains a number")
    End If
End Sub
```

Second approach

```
Sub Main(args As String())
    Dim sentence As String

    Dim found As Boolean
    Console.Write("Enter a sentence: ")
    sentence = Console.ReadLine()

    found = False
    For Each digit In "0123456789"
        If sentence.IndexOf(digit) > -1 Then
            found = True
            Exit For
        End If
    Next

    If found = True Then
        Console.WriteLine("The sentence contains a number")
    End If
End Sub
```

4. Solution

```
Console.WriteLine("Printing all integers from 1 to 100")
i = 1
Do While i < 101
    Console.WriteLine(i)
```

```
    i += 1
Loop
```

5. Solution

```
Console.WriteLine("Printing odd integers from 1 to 99")
i = 1
Do While Not (i > 100)
    Console.WriteLine(i)
    i += 2
Loop
```

6. Solution

```
Sub Main(args As String())
    Dim i, j As Integer

    For i = 1 To 4
        For j = 1 To 4
            Console.WriteLine(i & " x " & j & " = " & i * j)
        Next
    Next
End Sub
```

7. Solution

```
Sub Main(args As String())
    Dim i, j As Integer

    Console.Write(vbTab & "|" & vbTab)
    For i = 1 To 12
        Console.Write(i & vbTab)
    Next
    Console.WriteLine()

    For i = 1 To 12
        Console.Write("---------")
    Next
    Console.WriteLine()

    For i = 1 To 12
        Console.Write(i & vbTab & "|" & vbTab)
        For j = 1 To 12
            Console.Write(i * j & vbTab)
        Next
        Console.WriteLine()
    Next
End Sub
```

8. Solution

```
Sub Main(args As String())
```

```
    Dim i, j, n As Integer

    Console.Write("Enter an integer: ")
    n = Console.ReadLine()

    Console.Write(vbTab & "|" & vbTab)
    For i = 1 To n
        Console.Write(i & vbTab)
    Next
    Console.WriteLine()

    For i = 1 To n
        Console.Write("---------")
    Next
    Console.WriteLine()

    For i = 1 To n
        Console.Write(i & vbTab & "|" & vbTab)
        For j = 1 To n
            Console.Write(i * j & vbTab)
        Next
        Console.WriteLine()
    Next
End Sub
```

Chapter 21

21.2 Review Exercises

1. Solution

```
Sub Main(args As String())
    Dim i, total As Integer

    total = 0
    For i = 1 To 99 Step 2
        total += i
    Next

    Console.WriteLine(total)
End Sub
```

2. Solution

```
Sub Main(args As String())
    Dim n, total, i As Integer

    Console.Write("Enter N: ")
```

```
    n = Console.ReadLine()

    total = 0
    For i = 2 To 2 * n Step 2
        total += i
    Next

    Console.WriteLine(total)
End Sub
```

3. Solution

```
Sub Main(args As String())
    Dim count_pos, count_neg, total_pos, total_neg, i, x As Integer

    count_pos = 0
    count_neg = 0
    total_pos = 0
    total_neg = 0

    For i = 1 To 50
        Console.Write("Enter an integer: ")
        x = Console.ReadLine()
        If x > 0 Then
            count_pos += 1
            total_pos += x
        ElseIf x < 0 Then
            count_neg += 1
            total_neg += x
        End If
    Next

    If count_pos > 0 Then
        Console.WriteLine(total_pos / count_pos)
    End If

    If count_neg > 0 Then
        Console.WriteLine(total_neg / count_neg)
    End If
End Sub
```

4. Solution

```
Sub Main(args As String())
    Dim n, i, grade, total, count As Integer

    Console.Write("Enter total number of students: ")
    n = Console.ReadLine()

    total = 0
```

```
    count = 0
    For i = 1 To n
        Console.Write("Enter grade: ")
        grade = Console.ReadLine()
        If grade >= 90 And grade <= 100 Then
            total += grade
            count += 1
        End If
    Next

    If count > 0 Then
        Console.WriteLine(total / count)
    Else
        Console.WriteLine("There are no students that got an A")
    End If
End Sub
```

5. Solution

```
Sub Main(args As String())
    Dim count As Integer
    Dim total, x As Double

    total = 0
    count = 0
    Do
        Console.Write("Enter a number: ")
        x = Console.ReadLine()
        If x = 0 Then
            count += 1
        End If
        total += x
    Loop While total <= 3000

    Console.WriteLine(count)
End Sub
```

6. Solution

```
Sub Main(args As String())
    Dim answer As String
    Dim r, area As Double

    Do
        Console.Write("Enter the length of a radius of a circle: ")
        r = Console.ReadLine()

        area = 3.141 * r ^ 2
        Console.WriteLine("The area is: " & area)
```

```
        Console.Write("Would you like to repeat? ")
        answer = Console.ReadLine()
    Loop While answer.ToUpper() = "YES"
End Sub
```

7. Solution

```
Sub Main(args As String())
    Dim x As Long

    x = 1
    Do While x <= 1073741824
        Console.WriteLine(x)
        x *= 2
    Loop
End Sub
```

8. Solution

```
Sub Main(args As String())
    Dim i As Integer

    For i = 1 To 100
        Console.WriteLine(-i & vbCrLf & " " & i)
    Next
End Sub
```

9. Solution

First approach

```
Sub Main(args As String())
    Dim i As Integer
    Dim offset, value As Double

    value = 0
    For i = 0 To 7
        offset = 10 ^ i
        value += offset
        Console.WriteLine(value)
    Next
End Sub
```

Second approach

```
Sub Main(args As String())
    Dim value As String
    Dim i As Integer

    value = "1"
    For i = 0 To 7
        Console.WriteLine(value)
        value &= "1"
```

```
    Next
End Sub
```

10.Solution

```
Sub Main(args As String())
    Dim t, maximum, total As Double
    Dim i As Integer

    Console.Write("Enter temperature for day 1: ")
    t = Console.ReadLine()
    maximum = t
    total = t
    For i = 2 To 31
        Console.Write("Enter temperature for day " & i & ": ")
        t = Console.ReadLine()

        total += t
        If t > maximum Then
            maximum = t
        End If
    Next
    Console.WriteLine(total / 31 & " " & maximum)
End Sub
```

11.Solution

```
Sub Main(args As String())
    Dim level, maximum, minimum As Double
    Dim hour, min_hour, max_hour, i As Integer

    Console.Write("Enter level: ")
    level = Console.ReadLine()
    Console.Write("Enter hour: ")
    hour = Console.ReadLine()

    maximum = level
    minimum = level
    max_hour = hour
    min_hour = hour

    For i = 2 To 24
        Console.Write("Enter level: ")
        level = Console.ReadLine()
        Console.Write("Enter hour: ")
        hour = Console.ReadLine()

        If level > maximum Then
            maximum = level
```

```
            max_hour = hour
        End If

        If level < minimum Then
            minimum = level
            min_hour = hour
        End If
    Next
    Console.WriteLine(maximum & " " & max_hour & " " & minimum & " " & min_hour)
End Sub
```

12.Solution

```
Sub Main(args As String())
    Dim attempts As Integer
    Dim attempts_1st_player As Integer = 0, attempts_2nd_player As Integer = 0
    Dim guess, i, secret_number As Integer

    Dim rnd As New Random()

    For i = 1 To 2
        secret_number = rnd.Next(1, 101)

        attempts = 1
        Console.Write("Enter a guess: ")
        guess = Console.ReadLine()
        Do While guess <> secret_number
            If guess > secret_number Then
                Console.Write("Your guess is bigger than my secret number. Try again.")
            Else
                Console.Write("Your guess is smaller than my secret number. Try again.")
            End If
            attempts += 1
            Console.Write("Enter a guess: ")
            guess = Console.ReadLine()
        Loop

        Console.WriteLine("You found it!")
        Console.WriteLine("Attempts: " & attempts)

        If i = 0 Then
            attempts_1st_player = attempts
        Else
            attempts_2nd_player = attempts
        End If
    Next

    If attempts_1st_player < attempts_2nd_player Then
```

```
        Console.WriteLine("First Player Wins!")
    ElseIf attempts_1st_player > attempts_2nd_player Then
        Console.WriteLine("Second Player Wins!")
    Else
        Console.WriteLine("It's a draw")
    End If
End Sub
```

13.Solution

```
Sub Main(args As String())
    Dim gender As String
    Dim n, i, grade, total, total_a, count_a, total_b, count_b As Integer
    Dim total_a_boys, count_a_boys, count_cdef_girls As Integer

    Console.Write("Enter total number of students: ")
    n = Console.ReadLine()

    total = 0
    total_a = 0
    count_a = 0
    total_b = 0
    count_b = 0
    total_a_boys = 0
    count_a_boys = 0
    count_cdef_girls = 0

    For i = 1 To n
        Console.Write("Enter grade for student No " & i & ": ")
        grade = Console.ReadLine()

        Console.Write("Enter gender for student No " & i & ": ")
        gender = Console.ReadLine()

        If grade >= 90 And grade <= 100 Then
            total_a += grade
            count_a += 1
            If gender = "M" Then
                total_a_boys += grade
                count_a_boys += 1
            End If
        ElseIf grade >= 80 And grade <= 89 Then
            total_b += grade
            count_b += 1
        Else
            If gender = "F" Then
                count_cdef_girls += 1
            End If
```

```
        End If
        total += grade
    Next

    If count_a > 0 Then
        Console.Write("Average value of those who got an 'A': ")
        Console.WriteLine(total_a / count_a)
    End If

    If count_b > 0 Then
        Console.Write("Average value of those who got a 'B': ")
        Console.WriteLine(total_b / count_b)
    End If

    If count_a_boys > 0 Then
        Console.Write("Average value of boys who got an 'A': ")
        Console.WriteLine(total_a_boys / count_a_boys)
    End If

    Console.Write("Total number of girls that got less than 'B': ")
    Console.WriteLine(count_cdef_girls)

    Console.Write("Average grade of the whole class: ")
    Console.WriteLine(total / n)
End Sub
```

14.Solution

```
Sub Main(args As String())
    Dim answer As String
    Dim amount, discount As Double

    Do
        Console.Write("Enter amount: ")
        amount = Console.ReadLine()

        If amount < 20 Then
            discount = 0
        ElseIf amount < 50 Then
            discount = 3
        ElseIf amount < 100 Then
            discount = 5
        Else
            discount = 10
        End If

        Console.WriteLine("Discount: " & discount & "%")

        Console.Write("Would you like to repeat? ")
```

```
        answer = Console.ReadLine()
    Loop While answer.ToUpper() = "YES"
End Sub
```

Chapter 22

22.12 Review Exercises

1. Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Turtle.Forward(150)
    Turtle.RotateTo(-130)
    Turtle.Forward(50)

    Turtle.X = 0
    Turtle.Y = 150

    Turtle.RotateTo(130)
    Turtle.Forward(50)
End Sub
```

2. Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Turtle.Rotate(20)
    Turtle.Forward(100)
    Turtle.Rotate(90 - 20)
    Turtle.Forward(200)
    Turtle.Rotate(90 + 20)
    Turtle.Forward(100)
    Turtle.Rotate(90 - 20)
    Turtle.Forward(200)
End Sub
```

3. Solution

First approach

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Turtle.Rotate(30)
    Turtle.Forward(100)
    Turtle.Rotate(120)
    Turtle.Forward(100)
    Turtle.Rotate(60)
    Turtle.Forward(100)
    Turtle.Rotate(120)
    Turtle.Forward(100)
End Sub
```

Second approach

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Turtle.RotateTo(30)
    Turtle.Forward(100)
    Turtle.RotateTo(150)
    Turtle.Forward(100)
    Turtle.RotateTo(210)
    Turtle.Forward(100)
    Turtle.RotateTo(330)
    Turtle.Forward(100)
End Sub
```

4. Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Turtle.Rotate(45)
    Turtle.Forward(141)
    Turtle.Rotate(45)
    Turtle.Forward(100)
    Turtle.Rotate(45)
    Turtle.Forward(141)
    Turtle.Rotate(45 + 90)
    Turtle.Forward(300)
End Sub
```

5. Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim k, i As Integer

    Turtle.PenSize = 2
    For k = 1 To 2
        For i = 1 To 4
            Turtle.Forward(100)
            Turtle.Rotate(90)
        Next
        Turtle.X -= 100
    Next
End Sub
```

6. Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim m, n, i As Integer

    For m = 1 To 2
        For n = 1 To 2
            For i = 1 To 4
                Turtle.Forward(100)
                Turtle.Rotate(90)
```

```
            Next
            Turtle.X -= 150
        Next
        Turtle.X = 0
        Turtle.Y -= 150
    Next
End Sub
```

7. Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim size, length, height As Integer

    size = Interaction.InputBox("Enter pen size:")
    length = Interaction.InputBox("Enter length:")
    height = Interaction.InputBox("Enter heigth:")

    Turtle.PenSize = size
    Turtle.Forward(height)
    Turtle.Rotate(90)
    Turtle.Forward(length)
    Turtle.Rotate(90)
    Turtle.Forward(height)
    Turtle.Rotate(90)
    Turtle.Forward(length)
End Sub
```

8. Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim length As Integer = Interaction.InputBox("Enter length of the side:")

    Turtle.RotateTo(90)

    Turtle.Forward(length)
    Turtle.Rotate(-120)
    Turtle.Forward(length)
    Turtle.Rotate(-120)
    Turtle.Forward(length)
    Turtle.Rotate(-120)
    Turtle.Forward(length)
End Sub
```

9. Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim i As Integer

    For i = 1 To 12
        Turtle.Forward(100)
        Turtle.X = 0
```

```
        Turtle.Y = 0
        Turtle.Rotate(30)
    Next
End Sub
```

10.Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim k, i As Integer

    Turtle.PenSize = 2
    Turtle.RotateTo(90)
    For k = 1 To 180 Step 60
        For i = 1 To 5
            Turtle.Forward(150 + k)
            Turtle.Rotate(180 / 5 * 4)
        Next
        Turtle.X -= 30
        Turtle.Y += 10
    Next
End Sub
```

11.Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim k, i As Integer
    Turtle.PenSize = 2
    For k = 1 To 3
        For i = 1 To 4
            Turtle.Forward(100)
            Turtle.Rotate(90)
        Next
        Turtle.Rotate(-30)
    Next
End Sub
```

12.Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim k, i As Integer
    Turtle.PenSize = 2
    For k = 1 To 12
        For i = 1 To 4
            Turtle.Forward(100)
            Turtle.Rotate(90)
        Next
        Turtle.Rotate(-30)
    Next
End Sub
```

13.Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim k, i As Integer
    Turtle.PenSize = 2
    For k = 1 To 8
        For i = 1 To 4
            Turtle.Forward(100)
            Turtle.Rotate(90)
        Next
        Turtle.Rotate(-45)
    Next
End Sub
```

14.Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim i, k As Integer

    Turtle.PenSize = 1
    Turtle.PenColor = Color.Blue

    'Draw a blue rectangle
    Turtle.Forward(100)
    Turtle.Rotate(90)
    Turtle.Forward(200)
    Turtle.Rotate(90)
    Turtle.Forward(100)
    Turtle.Rotate(90)
    Turtle.Forward(200)

    'Move the turtle to the top left corner of the rectangle
    Turtle.Rotate(90)
    Turtle.PenUp()
    Turtle.Forward(100)
    Turtle.PenDown()

    'Draw the red roof
    Turtle.RotateTo(45)
    Turtle.PenColor = Color.Red
    Turtle.Forward(141)
    Turtle.Rotate(90)
    Turtle.Forward(141)

    'Draw the windows
    Turtle.PenColor = Color.Brown
    Turtle.X -= 180
    Turtle.Y = 50
```

```
    For k = 0 To 1
        Turtle.RotateTo(0)

        For i = 0 To 3
            Turtle.Forward(40)
            Turtle.Rotate(90)
        Next

        Turtle.RotateTo(0)
        Turtle.X += 20
        Turtle.Forward(40)

        Turtle.RotateTo(90)
        Turtle.X -= 20
        Turtle.Y = 70
        Turtle.Forward(40)

        Turtle.X += 80
        Turtle.Y -= 20
    Next

    'Draw the door
    Turtle.X -= 180
    Turtle.Y = 0
    Turtle.RotateTo(0)

    Turtle.Forward(70)
    Turtle.Rotate(90)
    Turtle.Forward(40)
    Turtle.Rotate(90)
    Turtle.Forward(70)
    Turtle.Rotate(90)
    Turtle.Forward(40)
End Sub
```

15.Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim m As Integer
    'Move to poll position
    Turtle.X = -300

    For m = 1 To 3
        'Here goes the code of the previous exersise

        Turtle.RotateTo(0)
        Turtle.X += 200
    Next
```

```
End Sub
```

The final program becomes

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim m As Integer
    'Move to poll position
    Turtle.X = -300

    For m = 1 To 3
        Dim i, k As Integer

        Turtle.PenSize = 1
        Turtle.PenColor = Color.Blue

        'Draw a blue rectangle
        Turtle.Forward(100)
        Turtle.Rotate(90)
        Turtle.Forward(200)
        Turtle.Rotate(90)
        Turtle.Forward(100)
        Turtle.Rotate(90)
        Turtle.Forward(200)

        'Move the turtle to the top left corner of the rectangle
        Turtle.Rotate(90)
        Turtle.PenUp()
        Turtle.Forward(100)
        Turtle.PenDown()

        'Draw the red roof
        Turtle.RotateTo(45)
        Turtle.PenColor = Color.Red
        Turtle.Forward(141)
        Turtle.Rotate(90)
        Turtle.Forward(141)

        'Draw the windows
        Turtle.PenColor = Color.Brown
        Turtle.X -= 180
        Turtle.Y = 50

        For k = 0 To 1
            Turtle.RotateTo(0)

            For i = 0 To 3
                Turtle.Forward(40)
                Turtle.Rotate(90)
            Next
```

```
            Turtle.RotateTo(0)
            Turtle.X += 20
            Turtle.Forward(40)

            Turtle.RotateTo(90)
            Turtle.X -= 20
            Turtle.Y = 70
            Turtle.Forward(40)

            Turtle.X += 80
            Turtle.Y -= 20
        Next

        'Draw the door
        Turtle.X -= 180
        Turtle.Y = 0
        Turtle.RotateTo(0)

        Turtle.Forward(70)
        Turtle.Rotate(90)
        Turtle.Forward(40)
        Turtle.Rotate(90)
        Turtle.Forward(70)
        Turtle.Rotate(90)
        Turtle.Forward(40)

        Turtle.RotateTo(0)
        Turtle.X += 200
    Next
End Sub
```

16.Solution

```
Private Sub Form1_Shown(sender As Object, e As EventArgs) Handles MyBase.Shown
    Dim i, j, sides, length As Integer
    Dim rnd As New Random()

    Turtle.PenSize = 2

    For i = 1 To 10
        'Pick random x, y values and move the turtle to that position
        Turtle.X = rnd.Next(-200, 201)
        Turtle.Y = rnd.Next(-200, 201)

        sides = rnd.Next(5, 10)   'Pick random number of sides
        length = rnd.Next(20, 50) 'Pick random length of sides

        'Draw the polygon
```

```
        For j = 1 To sides
            Turtle.Forward(length)
            Turtle.Rotate(360 / sides)
        Next
    Next
End Sub
```

Chapter 23

23.14 Review Questions: True/False

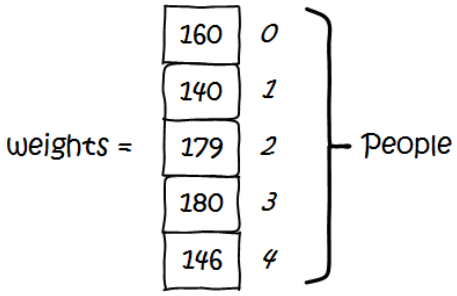
1. True
2. True
3. True
4. False
5. True
6. False
7. True
8. True
9. False
10. False
11. False
12. True
13. False
14. True
15. False
16. True
17. False
18. True
19. False
20. True
21. False
22. False
23. True
24. False
25. True
26. True
27. False
28. False
29. True
30. True
31. True
32. False
33. False
34. True
35. True
36. False
37. False
38. True
39. False
40. True
41. True
42. True
43. True
44. True
45. True
46. True
47. False

23.15 Review Questions: Multiple Choice

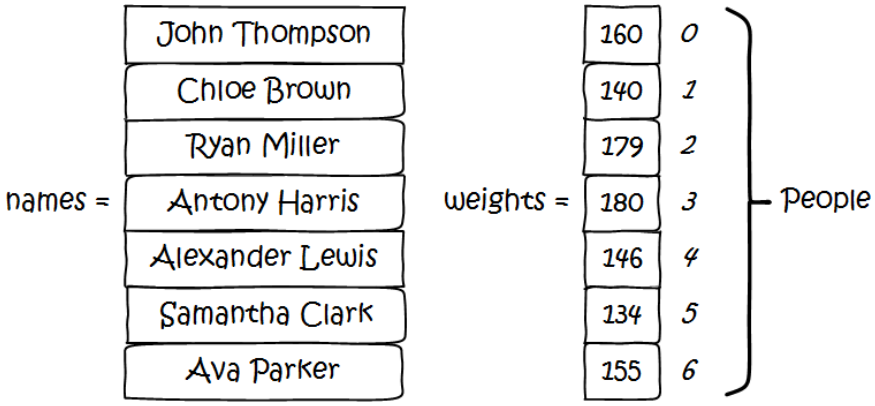
1. b	6. d	11. b	16. c
2. c	7. c	12. b	17. a
3. b	8. a	13. a	18. a
4. d	9. b	14. a	19. b
5. d	10. a	15. a	20. a

23.16 Review Exercises

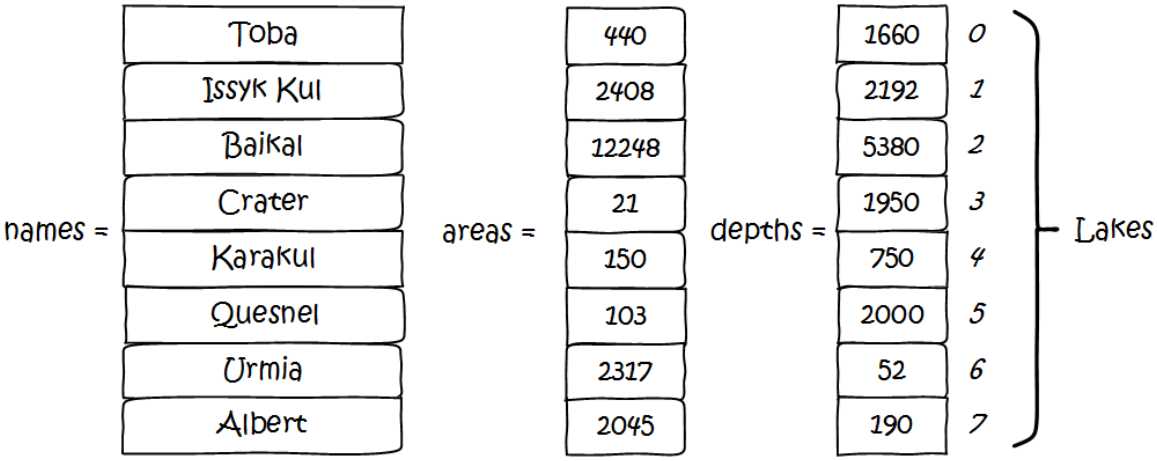
1. Solution



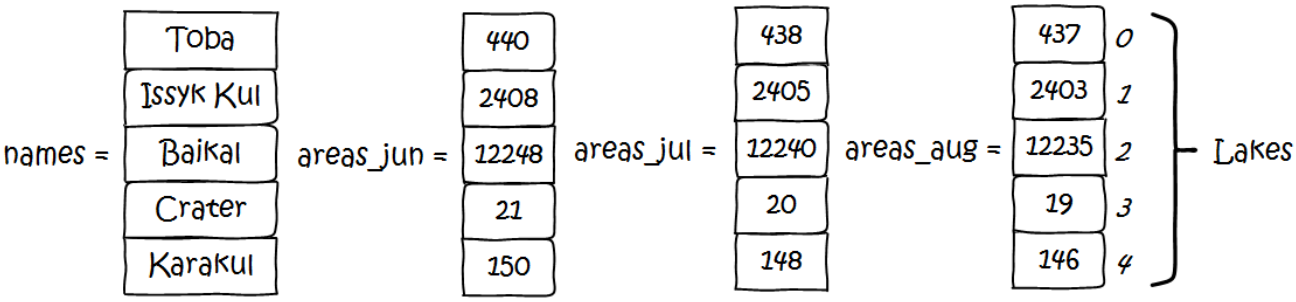
2. Solution



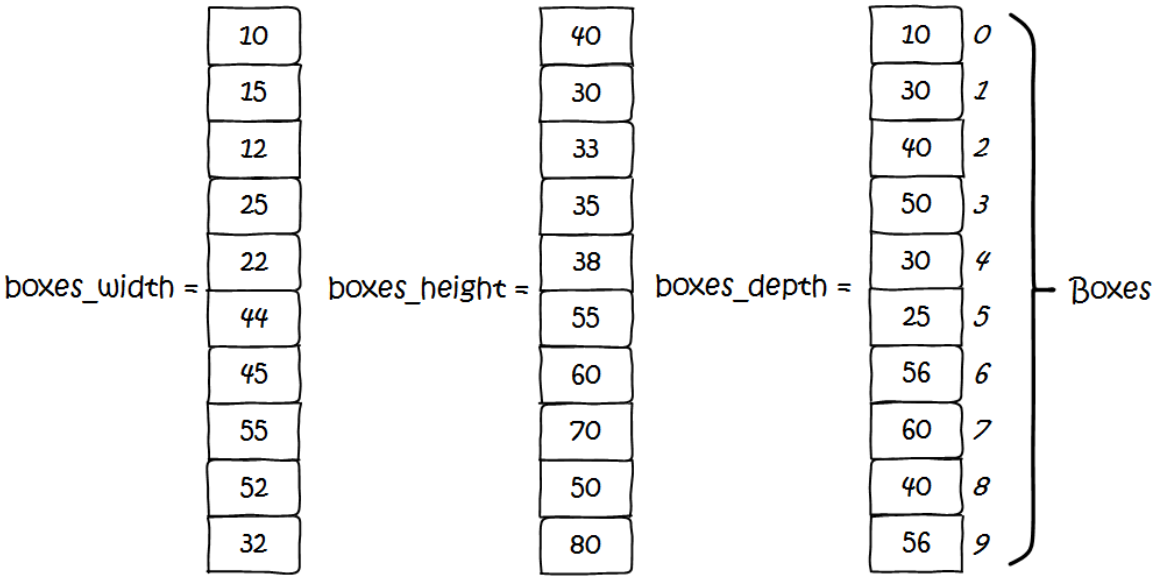
3. Solution



4. Solution



5. Solution



6. Solution

{16, 4, 1}

7. Solution

{4, 5, 11, 20, 10}

8. Solution

{18, 11, 46, 11, 11, 50}

9. Solution

{10, 22, 45, 67, 86, 19}

10.Solution

Navajo

Cherokee

Sioux

11.Solution

```
Const ELEMENTS = 100

Sub Main(args As String())
    Dim i As Integer
```

```
    Dim a(ELEMENTS - 1) As Double

    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a number: ")
        a(i) = Console.ReadLine()
    Next

    For i = 0 To ELEMENTS - 1
        Console.WriteLine(a(i) ^ 3)
    Next
End Sub
```

12.Solution

```
Const ELEMENTS = 80

Sub Main(args As String())
    Dim i As Integer

    Dim a(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a number: ")
        a(i) = Console.ReadLine()
    Next

    For i = 0 To ELEMENTS - 1
        a(i) = a(i) ^ 2
    Next

    For i = ELEMENTS - 1 To 0 Step -1
        Console.WriteLine(a(i))
    Next
End Sub
```

13.Solution

```
Const ELEMENTS = 50

Sub Main(args As String())
    Dim i As Integer

    Dim a(ELEMENTS - 1) As Integer
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter an integer: ")
        a(i) = Console.ReadLine()
    Next

    For Each element In a
        If element >= 10 Then
            Console.WriteLine(element)
```

```
        End If
    Next
End Sub
```

14.Solution

```
Const ELEMENTS = 30

Sub Main(args As String())
    Dim i As Integer
    Dim total As Double

    Dim a(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a number: ")
        a(i) = Console.ReadLine()
    Next

    total = 0
    For Each element In a
        If element > 0 Then
            total += element
        End If
    Next
    Console.WriteLine(total)
End Sub
```

15.Solution

```
Const ELEMENTS = 50

Sub Main(args As String())
    Dim i, total As Integer

    Dim a(ELEMENTS - 1) As Integer
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter an integer: ")
        a(i) = Console.ReadLine()
    Next

    total = 0
    For Each element In a
        If element >= 10 And element <= 99 Then
            total += element
        End If
    Next
    Console.WriteLine(total)
End Sub
```

16.Solution

```
Const ELEMENTS = 40

Sub Main(args As String())
    Dim i As Integer
    Dim total_pos, total_neg As Double

    Dim a(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a number: ")
        a(i) = Console.ReadLine()
    Next

    total_pos = 0
    total_neg = 0
    For Each element In a
        If element > 0 Then
            total_pos += element
        ElseIf element < 0 Then
            total_neg += element
        End If
    Next

    Console.WriteLine(total_pos & " " & total_neg)
End Sub
```

17.Solution

```
Const ELEMENTS = 20

Sub Main(args As String())
    Dim i As Integer
    Dim total As Double

    Dim a(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a number: ")
        a(i) = Console.ReadLine()
    Next

    total = 0
    For Each element In a
        total += element
    Next

    Console.WriteLine(total / ELEMENTS)
End Sub
```

18.Solution

```
Const ELEMENTS = 50

Sub Main(args As String())
    Dim i As Integer

    Dim a(ELEMENTS - 1) As Integer
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter an integer: ")
        a(i) = Console.ReadLine()
    Next

    For i = 0 To ELEMENTS - 1
        If a(i) < 20 Then
            Console.WriteLine(i)
        End If
    Next
End Sub
```

19.Solution

```
Const ELEMENTS = 60

Sub Main(args As String())
    Dim i As Integer

    Dim a(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a number: ")
        a(i) = Console.ReadLine()
    Next

    For i = 0 To ELEMENTS - 1 Step 2
        Console.WriteLine(a(i))
    Next
End Sub
```

20.Solution

```
Const ELEMENTS = 20

Sub Main(args As String())
    Dim i As Integer
    Dim total As Double

    Dim a(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a number: ")
        a(i) = Console.ReadLine()
```

```
    Next

    total = 0
    For i = 0 To ELEMENTS - 1 Step 2
        total += a(i)
    Next

    Console.WriteLine(total)
End Sub
```

21.Solution

First approach

```
Sub Main(args As String())
    Dim i As Integer
    Dim a(99) As Integer
    For i = 0 To 99
        a(i) = i + 1
    Next
End Sub
```

Second approach

```
Sub Main(args As String())
    Dim i As Integer
    Dim a(99) As Integer
    For i = 1 To 100
        a(i - 1) = i
    Next
End Sub
```

22.Solution

First approach

```
Sub Main(args As String())
    Dim i As Integer
    Dim a(99) As Integer
    For i = 0 To 99
        a(i) = 2 * (i + 1)
    Next
End Sub
```

Second approach

```
Sub Main(args As String())
    Dim i As Integer
    Dim a(99) As Integer
    For i = 1 To 100
        a(i - 1) = 2 * i
    Next
End Sub
```

23.Solution

```
Sub Main(args As String())
    Dim i, n As Integer

    Console.Write("Enter N: ")
    n = Console.ReadLine()

    Dim a(n - 1) As Double
    For i = 1 To n
        a(i - 1) = i ^ 2
    Next

    For Each element In a
        Console.WriteLine(element)
    Next
End Sub
```

24.Solution

```
Const ELEMENTS = 10

Sub Main(args As String())
    Dim i As Integer

    Dim a(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a number: ")
        a(i) = Console.ReadLine()
    Next

    For i = 0 To ELEMENTS - 1
        If a(i) = Fix(a(i)) Then
            Console.WriteLine(i)
        End If
    Next
End Sub
```

25.Solution

```
Const ELEMENTS = 50

Sub Main(args As String())
    Dim i, count As Integer

    Dim a(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a number: ")
        a(i) = Console.ReadLine()
    Next
```

```
    count = 0
    For i = 0 To ELEMENTS - 1
        If a(i) < 0 Then
            count += 1
        End If
    Next

    Console.WriteLine(count)
End Sub
```

26.Solution

```
Const ELEMENTS = 50

Sub Main(args As String())
    Dim i As Integer

    Dim words(ELEMENTS - 1) As String
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a word: ")
        words(i) = Console.ReadLine()
    Next

    For Each word In words
        If word.Length >= 10 Then
            Console.WriteLine(word)
        End If
    Next
End Sub
```

27.Solution

```
Const ELEMENTS = 30

Sub Main(args As String())
    Dim i As Integer

    Dim words(ELEMENTS - 1) As String
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a word: ")
        words(i) = Console.ReadLine()
    Next

    Dim length_limits() As Integer = {5, 10, 20}

    For Each length_limit In length_limits
        For Each word In words
            If word.Length < length_limit Then
                Console.WriteLine(word)
            End If
```

```
        Next
    Next
End Sub
```

28. Solution

```
Const ELEMENTS = 40

Sub Main(args As String())
    Dim i, count As Integer

    Dim words(ELEMENTS - 1) As String
    For i = 0 To ELEMENTS - 1
        Console.Write("Enter a word: ")
        words(i) = Console.ReadLine()
    Next

    For Each word In words
        count = 0
        For Each letter In word
            If letter = "w" Then
                count += 1
            End If

            If count = 2 Then
                Console.WriteLine(word)
                Exit For
            End If
        Next
    Next
End Sub
```

Chapter 24

24.5 Review Questions: True/False

1. False
2. False
3. True
4. True

24.6 Review Exercises

1. Solution

```
Const ELEMENTS_OF_A = 50
Const ELEMENTS_OF_NEW = ELEMENTS_OF_A - 2

Sub Main(args As String())
    Dim i As Integer

    Dim a(ELEMENTS_OF_A - 1) As Double
```

```
    For i = 0 To ELEMENTS_OF_A - 1
        Console.Write("Enter a number: ")
        a(i) = Console.ReadLine()
    Next

    Dim new_arr(ELEMENTS_OF_NEW - 1) As Double
    For i = 0 To ELEMENTS_OF_NEW - 1
        new_arr(i) = (a(i) + a(i + 1) + a(i + 2)) / 3
    Next

    For i = 0 To ELEMENTS_OF_NEW - 1
        Console.WriteLine(new_arr(i))
    Next
End Sub
```

2. Solution

```
Const ELEMENTS = 15

Sub Main(args As String())
    Dim i As Integer
    Dim minimum As Double

    Dim a(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        Console.Write("A - Enter a number: ")
        a(i) = Console.ReadLine()
    Next

    Dim b(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        Console.Write("B - Enter a number: ")
        b(i) = Console.ReadLine()
    Next

    Dim c(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        Console.Write("C - Enter a number: ")
        c(i) = Console.ReadLine()
    Next

    Dim new_arr(ELEMENTS - 1) As Double
    For i = 0 To ELEMENTS - 1
        minimum = a(i)
        If b(i) < minimum Then
            minimum = b(i)
        End If
        If c(i) < minimum Then
```

```
            minimum = c(i)
        End If
        new_arr(i) = minimum
    Next

    For i = 0 To ELEMENTS - 1
        Console.WriteLine(new_arr(i))
    Next
End Sub
```

3. Solution

```
Const MOUNTAINS = 30

Sub Main(args As String())
    Dim i, index_of_max, index_of_min As Integer
    Dim maximum, minimum As Double

    Dim names(MOUNTAINS - 1) As String
    Dim heights(MOUNTAINS - 1) As Double
    Dim countries(MOUNTAINS - 1) As String
    For i = 0 To MOUNTAINS - 1
        names(i) = Console.ReadLine()
        heights(i) = Console.ReadLine()
        countries(i) = Console.ReadLine()
    Next

    maximum = heights(0)
    index_of_max = 0
    minimum = heights(0)
    index_of_min = 0
    For i = 1 To MOUNTAINS - 1
        If heights(i) > maximum Then
            maximum = heights(i)
            index_of_max = i
        End If
        If heights(i) < minimum Then
            minimum = heights(i)
            index_of_min = i
        End If
    Next

    Console.WriteLine(heights(index_of_max) & ", " & names(index_of_max) & ", " &
                      countries(index_of_max))
    Console.WriteLine(heights(index_of_min) & ", " & names(index_of_min) & ", " &
                      countries(index_of_min))
End Sub
```

4. Solution

```
Const CLASS1 = 20
Const CLASS2 = 25

Sub Main(args As String())
    Dim i As Integer
    Dim needle As String
    Dim found As Boolean

    Console.WriteLine("Class A")
    Dim names1(CLASS1 - 1) As String
    For i = 0 To CLASS1 - 1
        Console.Write("Enter name: ")
        names1(i) = Console.ReadLine()
    Next

    Console.WriteLine("Class B")
    Dim names2(CLASS2 - 1) As String
    For i = 0 To CLASS2 - 1
        Console.Write("Enter name: ")
        names2(i) = Console.ReadLine()
    Next

    Console.Write("Enter a name to search: ")
    needle = Console.ReadLine()

    found = False
    For Each name In names1
        If name = needle Then
            found = True
            Exit For
        End If
    Next

    If found = True Then
        Console.WriteLine("Student found in class No 1")
    Else
        found = False
        For Each name In names2
            If name = needle Then
                found = True
                Exit For
            End If
        Next

        If found = True Then
```

```
            Console.WriteLine("Student found in class No 2")
        Else
            Console.WriteLine("Student not found in either class")
        End If
    End If
End Sub
```

5. Solution

```
Console.Write("Enter username: ")
usr = Console.ReadLine()
Console.Write("Enter password: ")
pwd = Console.ReadLine()

found = False
For i = 0 To 99
    If usernames(i) = usr Then
        found = True
        Exit For
    End If
Next

If found = True Then
    If usernames(i) = usr And passwords(i) = pwd Then
        Console.WriteLine("Login OK!")
    Else
        Console.WriteLine("Login Failed!")
    End If
Else
    Console.WriteLine("Login Failed!")
End If
```

6. Solution

```
Console.Write("Enter a value to search: ")
needle = Console.ReadLine()

found = False
For i = 0 To 999
    If SSNs(i) = needle Then
        found = True
        Console.WriteLine(SSNs(i) & " " & names(i))
        Exit For
    End If
Next

If found = False Then
    For i = 0 To 999
```

```
        If names(i) = needle Then
            found = True
            Console.WriteLine(SSNs(i) & " " & names(i))
        End If
    Next
End If

If found = False Then
    Console.WriteLine("This value does not exist")
End If
```

7. Solution

```
Const STUDENTS = 12

Sub Main(args As String())
    Dim i As Integer
    Dim found As Boolean

    Dim grades1(STUDENTS - 1) As Integer
    Dim grades2(STUDENTS - 1) As Integer
    Dim grades3(STUDENTS - 1) As Integer
    For i = 0 To STUDENTS - 1
        grades1(i) = Console.ReadLine()
        grades2(i) = Console.ReadLine()
        grades3(i) = Console.ReadLine()
    Next

    found = False
    For i = 0 To STUDENTS - 1
        If (grades1(i) + grades2(i) + grades3(i)) / 3 < 70 Then
            found = True
            Exit For
        End If
    Next

    If found = True Then
        Console.Write("There is at least one student has an average value below 70")
    End If
End Sub
```

8. Solution

```
Const STUDENTS = 15

Sub Main(args As String())
    Dim i As Integer
    Dim average As Double

    Dim grades1(STUDENTS - 1) As Integer
```

```
    Dim grades2(STUDENTS - 1) As Integer
    For i = 0 To STUDENTS - 1
        grades1(i) = Console.ReadLine()
        grades2(i) = Console.ReadLine()
    Next

    For i = 0 To STUDENTS - 1
        Console.WriteLine("Student No " & i + 1 & ": ")

        average = (grades1(i) + grades2(i)) / 2

        If average < 60 Then
            Console.WriteLine("E/F")
        ElseIf average < 70 Then
            Console.WriteLine("D")
        ElseIf average < 80 Then
            Console.WriteLine("C")
        ElseIf average < 90 Then
            Console.WriteLine("B")
        Else
            Console.WriteLine("A")
        End If
    Next
End Sub
```

9. Solution

```
Const PLAYERS = 15

Sub Main(args As String())
    Dim i, total As Integer

    Dim points_match1(PLAYERS - 1) As Integer
    Dim points_match2(PLAYERS - 1) As Integer
    Dim points_match3(PLAYERS - 1) As Integer
    Dim points_match4(PLAYERS - 1) As Integer
    For i = 0 To PLAYERS - 1
        points_match1(i) = Console.ReadLine()
        points_match2(i) = Console.ReadLine()
        points_match3(i) = Console.ReadLine()
        points_match4(i) = Console.ReadLine()
    Next

    For i = 0 To PLAYERS - 1
        Console.WriteLine("Player No " & i + 1)
        total = points_match1(i) + points_match2(i) + points_match3(i) + points_match4(i)
        Console.WriteLine(total)
    Next
```

```
End Sub
```

10.Solution

```
Const HOURS = 24

Sub Main(args As String())
    Dim i As Integer
    Dim average As Double

    Dim t_city1(HOURS - 1) As Double
    Dim t_city2(HOURS - 1) As Double
    Dim t_city3(HOURS - 1) As Double
    For i = 0 To HOURS - 1
        t_city1(i) = Console.ReadLine()
        t_city2(i) = Console.ReadLine()
        t_city3(i) = Console.ReadLine()
    Next

    For i = 0 To HOURS - 1
        average = (t_city1(i) + t_city2(i) + t_city3(i)) / 3
        If average < 10 Then
            Console.WriteLine("Hour: " & i + 1)
        End If
    Next
End Sub
```

11.Solution

```
Const STUDENTS = 12

Sub Main(args As String())
    Dim i As Integer

    Dim names(STUDENTS - 1) As String
    Dim grd_lesson1(STUDENTS - 1) As Integer
    Dim grd_lesson2(STUDENTS - 1) As Integer
    For i = 0 To STUDENTS - 1
        names(i) = Console.ReadLine()
        grd_lesson1(i) = Console.ReadLine()
        grd_lesson2(i) = Console.ReadLine()
    Next

    'Create array average
    Dim average(STUDENTS - 1) As Double
    For i = 0 To STUDENTS - 1
        average(i) = (grd_lesson1(i) + grd_lesson2(i)) / 2
    Next

    For i = 0 To STUDENTS - 1
```

```
        Console.WriteLine(names(i) & " " & average(i))
    Next

    For i = 0 To STUDENTS - 1
        If average(i) < 60 Then
            Console.WriteLine(names(i))
        End If
    Next

    For i = 0 To STUDENTS - 1
        If average(i) > 89 Then
            Console.WriteLine(names(i) & " Bravo!")
        End If
    Next
End Sub
```

12.Solution

```
Const ARTISTS = 15

Sub Main(args As String())
    Dim i, minimum As Integer

    Dim artist_names(ARTISTS - 1) As String
    Dim song_titles(ARTISTS - 1) As String
    Dim scoreA(ARTISTS - 1) As Integer
    Dim scoreB(ARTISTS - 1) As Integer
    Dim scoreC(ARTISTS - 1) As Integer

    For i = 0 To ARTISTS - 1
        Console.Write("Name for artist No " & i + 1 & ": ")
        artist_names(i) = Console.ReadLine()

        Console.Write("Song title for artist " & artist_names(i) & ": ")
        song_titles(i) = Console.ReadLine()

        Console.WriteLine("Score for artist " & artist_names(i))

        Console.Write(" gotten from judge A: ")
        scoreA(i) = Console.ReadLine()

        Console.Write(" gotten from judge B: ")
        scoreB(i) = Console.ReadLine()

        Console.Write(" gotten from judge C: ")
        scoreC(i) = Console.ReadLine()
    Next

    Dim total(ARTISTS - 1) As Integer
```

```
    For i = 0 To ARTISTS - 1
        minimum = scoreA(i)
        If scoreB(i) < minimum Then
            minimum = scoreB(i)
        End If
        If scoreC(i) < minimum Then
            minimum = scoreC(i)
        End If

        total(i) = scoreA(i) + scoreB(i) + scoreC(i) - minimum
    Next

    For i = 0 To ARTISTS - 1
        Console.WriteLine(artist_names(i) & " " & song_titles(i) & " " & total(i))
    Next
End Sub
```

13.Solution

```
Const CITIZENS = 20

Sub Main(args As String())
    Dim prod_name1, prod_name2 As String
    Dim i, count_A As Integer

    Dim answers1(CITIZENS - 1) As String
    Dim answers2(CITIZENS - 1) As String

    Console.Write("Enter Product Name 1: ")
    prod_name1 = Console.ReadLine()
    For i = 0 To CITIZENS - 1
        Console.Write("Enter score for product " & prod_name1 & ": ")
        answers1(i) = Console.ReadLine()
    Next

    Console.Write("Enter Product Name 2: ")
    prod_name2 = Console.ReadLine()
    For i = 0 To CITIZENS - 1
        Console.Write("Enter score for product " & prod_name2 & ": ")
        answers2(i) = Console.ReadLine()
    Next

    count_A = 0
    For i = 0 To CITIZENS - 1
        If answers1(i) = "A" Then
            count_A += 1
        End If
```

```
    Next
    Console.WriteLine(prod_name1 & " " & count_A)

    count_A = 0
    For i = 0 To CITIZENS - 1
        If answers2(i) = "A" Then
            count_A += 1
        End If
    Next
    Conso
```

14.Solution

```
Sub Main(args As String())
    Dim msg As String

    Dim morseAlphabet As New Dictionary(Of String, String) From {
        {"A", ".-"},
        {"B", "-..."},
        {"C", "-.-."},
        {"D", "-.."},
        {"E", "."},
        {"F", "..-."},
        {"G", "--."},
        {"H", "...."},
        {"I", ".."},
        {"J", ".---"},
        {"K", "-.-"},
        {"L", ".-.."},
        {"M", "--"},
        {"N", "-."},
        {"O", "---"},
        {"P", ".--."},
        {"Q", "--.-"},
        {"R", ".-."},
        {"S", "..."},
        {"T", "-"},
        {"U", "..-"},
        {"V", "...-"},
        {"W", ".--"},
        {"X", "-..-"},
        {"Y", "-.--"},
        {"Z", "--.."},
        {" ", "/"}
    }

    Console.Write("Enter an English message: ")
```

```
    msg = Console.ReadLine()

    For Each character In msg
        Console.Write(morseAlphabet(character.ToString().ToUpper()))
    Next
End Sub
```

15.Solution

```
Sub Main(args As String())
    Dim word, letter As String
    Dim i, random_index, wrong_guesses As Integer

    Dim rnd As New Random()

    Dim words() As String = {"compiler", "interpreter", "error", "variable",
                             "operator", "computer", "programmer", "algorithm"}

    'Randomly choose a word
    random_index = rnd.Next(words.Length)  'Alternatively, these can be written as:
    word = words(random_index)             'word = words(rnd.Next(words.Length))

    wrong_guesses = 0

    'Create array results
    Dim results(word.Length - 1) As String
    For i = 0 To results.Length - 1
        results(i) = "_"
    Next

    'The method Contains() returns True when the array results contains the underscore
    'character within its elements.
    Do While results.Contains("_") And wrong_guesses < 6
        'Display results
        For Each x In results
            Console.Write(x & "  ")
        Next

        Console.Write("Enter a letter: ")
        letter = Console.ReadLine()

        If word.IndexOf(letter) > -1 Then
            'Replace corresponding underscores with letter in array results
            For i = 0 To word.Length - 1
                If letter = word(i) Then
                    results(i) = letter
                End If
            Next
        Else
```

```
            wrong_guesses += 1
            Console.WriteLine("This letter does not exist in clue word!")
        End If
    Loop

    If wrong_guesses < 6 Then
        Console.WriteLine("Congratulations, you found it!")
    Else
        Console.WriteLine("Game over!!!")
    End If
End Sub
```

16.Solution

```
Sub Main(args As String())
    Dim word, letter As String
    Dim i, player, wrong_guesses As Integer

    Dim rnd As New Random()

    Dim words() As String = {"compiler", "interpreter", "error", "variable",
                             "operator", "computer", "programmer", "algorithm"}

    Dim wrong_guesses1 As Integer = 0, wrong_guesses2 As Integer = 0

    For player = 1 To 2
        word = words(rnd.Next(words.Length))     'Randomly choose a word

        wrong_guesses = 0

        'Create array results
        Dim results(word.Length - 1) As String
        For i = 0 To results.Length - 1
            results(i) = "_"
        Next

        Do While results.Contains("_") And wrong_guesses < 6
            'Display results
            For Each x In results
                Console.Write(x & "  ")
            Next

            Console.Write("Player No: " & player & " - Enter a letter: ")
            letter = Console.ReadLine()

            If word.IndexOf(letter) > -1 Then
                'Replace corresponding underscores with letter in array results
                For i = 0 To word.Length - 1
                    If letter = word(i).ToString() Then
```

```
                        results(i) = letter
                    End If
                Next
            Else
                wrong_guesses += 1
                Console.WriteLine("This letter does not exist in clue word!")
            End If
        Loop

        Console.Write("Player No:" & player & " - ")
        If wrong_guesses < 6 Then
            Console.WriteLine("Congratulations, you found it!")
        Else
            Console.WriteLine("Game over!!!")
        End If

        If player = 1 Then
            wrong_guesses1 = wrong_guesses
        Else
            wrong_guesses2 = wrong_guesses
        End If
    Next

    'Display the winner
    If wrong_guesses1 < wrong_guesses2 Then
        Console.WriteLine("Winner: Player 1")
    ElseIf wrong_guesses2 < wrong_guesses1 Then
        Console.WriteLine("Winner: Player 2")
    Else
        Console.WriteLine("It's a tie!")
    End If
End Sub
```

Chapter 25

25.4 Review Questions: True/False

1. True
2. True
3. False
4. False
5. True
6. True
7. True
8. True
9. True
10. False
11. True

Chapter 26

26.11 Review Questions: True/False

1. True
2. True
3. False
4. True
5. True
6. True
7. False
8. True
9. True
10. True
11. False
12. True
13. True
14. True
15. True
16. True
17. False
18. False
19. False
20. False
21. True
22. True
23. True
24. False

26.12 Review Exercises

1. Solution

```
Function find_max(a As Integer, b As Integer)
    Dim maximum As Integer

    If a > b Then
        maximum = a
    Else
        maximum = b
    End If
    Return maximum
End Function
```

2. Solution

It displays:

3 is positive

-7 is negative or zero

-9 is negative or zero

0 is negative or zero

4 is positive

3. Solution

```
Function find_sum(a As Double, b As Double, c As Double) As Double
    Return a + b + c
End Function
```

4. Solution

```
Function find_avg(a As Double, b As Double, c As Double, d As Double) As Double
    Return (a + b + c + d) / 4
End Function
```

5. Solution

```
Function maximum(a As Double, b As Double, c As Double) As Double
    Dim m As Double

    m = a
    If b > m Then
```

```
        m = b
    End If
    If c > m Then
        m = c
    End If

    Console.WriteLine(m)
End Function
```

6. Solution

```
Function find_min(a As Double, b As Double) As Double
    Dim minimum As Double
    minimum = a
    If b < minimum Then
        minimum = b
    End If
    Return minimum
End Function

Sub Main(args As String())
    Dim x1, x2, x3, x4, temp1, temp2 As Double

    Console.WriteLine("Enter four numbers: ")
    x1 = Console.ReadLine()
    x2 = Console.ReadLine()
    x3 = Console.ReadLine()
    x4 = Console.ReadLine()

    'Display lowest value as follows (1st approach)
    temp1 = find_min(x1, x2)
    temp2 = find_min(x3, x4)
    Console.WriteLine(find_min(temp1, temp2))

    'Display lowest value as follows (2nd approach)
    Console.WriteLine(find_min(find_min(x1, x2), find_min(x3, x4)))
End Sub
```

7. Solution

```
Function get_input() As Boolean
    Dim answer As String
    Console.Write("Enter Yes or No: ")
    answer = Console.ReadLine()
    If answer.ToUpper() = "YES" Then
        Return True
    Else
        Return False
    End If
```

```
End Function

Function find_area(b As Double, h As Double) As Double
    Return b * h
End Function

Sub Main(args As String())
    Dim b, h As Double
    Dim answer As Boolean

    Do
        Console.Write("Enter the base of the parallelogram: ")
        b = Console.ReadLine()

        Console.Write("Enter the height of the parallelogram: ")
        h = Console.ReadLine()

        Console.WriteLine("Area = " & find_area(b, h))

        Console.WriteLine("Would you like to repeat? ")
        answer = get_input()    'Or you can write...
    Loop While answer = True    'Loop While get_input()
End Sub
```

Chapter 27

27.2 Review Exercises

1. Solution

```
Function Kelvin_to_Fahrenheit(kelvin As Double) As Double
    Return 1.8 * kelvin - 459.67
End Function

Function Kelvin_to_Celsius(kelvin As Double) As Double
    Return kelvin - 273.15
End Function

Sub Main(args As String())
    Dim k As Double

    Console.Write("Enter a temperature in degrees Kelvin: ")
    k = Console.ReadLine()

    Console.WriteLine("Fahrenheit: " & Kelvin_to_Fahrenheit(k))
    Console.Write("Celsius: " & Kelvin_to_Celsius(k))
End Sub
```

2. Solution

```
Function num_of_days(month As Integer) As Integer
    Dim days As Integer

    If month = 4 Or month = 6 Or month = 9 Or month = 11 Then
        days = 30
    ElseIf month = 2 Then
        days = 28
    Else
        days = 31
    End If
    Return days
End Function

Sub Main(args As String())
    Dim i, x, y, total As Integer

    Console.Write("Enter a month: ")
    x = Console.ReadLine()

    Console.Write("Enter a second month: ")
    y = Console.ReadLine()

    total = 0
    For i = x To y
        total += num_of_days(i)
    Next

    Console.WriteLine(total)
End Sub
```

3. Solution

```
Function dice() As Integer
    Dim rnd As New Random()
    Return rnd.Next(1, 7)
End Function

Sub Main(args As String())
    'Main code starts here
    Dim i, dice1, dice2, player As Integer

    Dim names(1) As String
    Console.Write("Player 1 enter name: ")
    names(0) = Console.ReadLine()
    Console.Write("Player 2 enter name: ")
    names(1) = Console.ReadLine()
```

```
    Dim total(1) As Integer

    For player = 0 To 1
        total(player) = 0
        For i = 1 To 10
            Console.Write(names(player))
            Console.WriteLine(", hit the Enter key to roll the dice!")
            Console.ReadLine()

            dice1 = dice()
            dice2 = dice()
            Console.WriteLine(dice1 & " " & dice2)
            total(player) += dice1 + dice2
        Next
    Next

    If total(0) = total(1) Then
        Console.WriteLine("Tie!")
    ElseIf total(0) > total(1) Then
        Console.WriteLine(names(0) & " wins!")
    Else
        Console.WriteLine(names(1) & " wins!")
    End If
End Sub
```

4. Solution

```
Sub bmi(w As Double, h As Double)
    Dim b As Double

    b = w * 703 / h ^ 2
    If b < 16 Then
        Console.WriteLine("You must add weight.")
    ElseIf b < 18.5 Then
        Console.WriteLine("You should add some weight.")
    ElseIf b < 25 Then
        Console.WriteLine("Maintain your weight.")
    ElseIf b < 30 Then
        Console.WriteLine("You should lose some weight.")
    Else
        Console.WriteLine("You must lose weight.")
    End If
End Sub

Sub Main(args As String())
    Dim height, weight As Double
    Dim age As Integer
```

```
    Console.Write("Enter your weight (in pounds): ")
    weight = Console.ReadLine()

    Console.WriteLine("Enter your age: ")
    age = Console.ReadLine()

    Console.WriteLine("Enter your height (in inches): ")
    height = Console.ReadLine()

    If age < 18 Then
        Console.WriteLine("I can't calculate your BMI. You must be adult!")
    Else
        bmi(weight, height)
    End If
End Sub
```

5. Solution

```
Const CARS = 40
Const GAS = 1
Const DIESEL = 2
Const HYBRID = 3

Function get_choice() As Integer
    Dim choice As Integer

    Console.WriteLine("1. Gas")
    Console.WriteLine("2. Diesel")
    Console.WriteLine("3. Hybrid")
    Console.Write("Enter type of the car: ")
    choice = Console.ReadLine()
    Return choice
End Function

Function get_days() As Integer
    Dim days As Integer

    Console.Write("Enter total number of rental days: ")
    days = Console.ReadLine()
    Return days
End Function

Function get_charge(car_type As Integer, rental_days As Integer) As Double
    Dim charge As Double

    If car_type = GAS Then
        If rental_days <= 5 Then
            charge = rental_days * 24
        Else
```

```
            charge = rental_days * 22
        End If
    ElseIf car_type = DIESEL Then
        If rental_days <= 5 Then
            charge = rental_days * 28
        Else
            charge = rental_days * 25
        End If
    Else
        If rental_days <= 5 Then
            charge = rental_days * 30
        Else
            charge = rental_days * 28
        End If
    End If
    Return charge
End Function

Sub Main(args As String())
    'Main code starts here
    Dim i, count As Integer
    Dim charge, total As Double

    Dim rented_car_types(CARS - 1) As Integer
    Dim rented_days(CARS - 1) As Integer

    For i = 0 To CARS - 1
        rented_car_types(i) = get_choice()
        rented_days(i) = get_days()
    Next

    total = 0
    For i = 0 To CARS - 1
        charge = get_charge(rented_car_types(i), rented_days(i))
        Console.WriteLine("Amount to pay, car No " & i + 1 & ": " & charge)
        total += charge
    Next

    count = 0
    For i = 0 To CARS - 1
        If rented_car_types(i) = HYBRID Then
            count += 1
        End If
    Next
    Console.WriteLine("Hybrids rented: " & count)
    Console.WriteLine("Total profit: " & total)
```

```
End Sub
```

Chapter 28

28.8 Review Questions: True/False

1. False
2. True
3. True
4. False
5. False
6. False
7. False
8. True
9. True
10. False
11. True
12. True
13. True
14. False
15. True
16. False
17. False

28.9 Review Exercises

1. Solution

```
Class Trigonometry
    Public Function square_area(side As Double) As Double
        Return side * side
    End Function

    Public Function rectangle_area(b As Double, h As Double) As Double
        Return b * h
    End Function

    Public Function triangle_area(b As Double, h As Double) As Double
        Return b * h / 2
    End Function
End Class

Sub Main(args As String())
    Dim sqr_side, rctngl_base, rctngl_height, trngl_base, trngl_height As Double

    Dim tr As New Trigonometry()

    Console.Write("Enter square side: ")
    sqr_side = Console.ReadLine()

    Console.Write("Enter rectangle base: ")
    rctngl_base = Console.ReadLine()
    Console.Write("Enter rectangle height: ")
    rctngl_height = Console.ReadLine()

    Console.Write("Enter triangle base: ")
    trngl_base = Console.ReadLine()
    Console.Write("Enter triangle height: ")
    trngl_height = Console.ReadLine()
```

```
    Console.WriteLine(tr.square_area(sqr_side))
    Console.WriteLine(tr.rectangle_area(rctngl_base, rctngl_height))
    Console.WriteLine(tr.triangle_area(trngl_base, trngl_height))
End Sub
```

2. Solution

```
Class Pet
    Public kind As String
    Public legs_number As Integer

    Public Sub start_running()
        Console.WriteLine("Pet is running")
    End Sub

    Public Sub stop_running()
        Console.WriteLine("Pet stopped")
    End Sub
End Class

Sub Main(args As String())
    Dim pet1 As New Pet()
    pet1.kind = "dog"
    pet1.legs_number = 4

    Dim pet2 As New Pet()
    pet2.kind = "monkey"
    pet2.legs_number = 2

    pet1.start_running()
    pet2.start_running()
    pet1.stop_running()

End Sub
```

3. Solution

```
Class Pet
    Private _kind As String
    Private _legs_number As Integer

    'Define the constructor
    Public Sub New(k As String, l As Integer)
        Me.Kind = k          'Initial value for the property Kind
        Me.Legs_number = l   'Initial value for the property Legs_number
    End Sub

    'Defines a public property
    Public Property Kind As String
```

```
        'Define the getter
        Get
            Return Me._kind
        End Get
        'Define the setter
        Set
            If value <> "" Then
                Me._kind = value
            Else
                Throw New Exception("Cannot be empty")
            End If
        End Set
    End Property

    'Defines a public property
    Public Property Legs_number As Integer
        'Define the getter
        Get
            Return Me._legs_number
        End Get
        'Define the setter
        Set
            If value >= 0 Then
                Me._legs_number = value
            Else
                Throw New Exception("Cannot be negative")
            End If
        End Set
    End Property

    Public Sub start_running()
        Console.WriteLine("Pet is running")
    End Sub

    Public Sub stop_running()
        Console.WriteLine("Pet stopped")
    End Sub
End Class

Sub Main(args As String())
    Dim pet1 As New Pet("dog", 4)

    pet1.start_running()
    pet1.stop_running()

    pet1.Kind = ""          'This will throw an error
```

```
    pet1.Legs_number = -3  'This will throw an error
End Sub
```

4. Solution

```
Const BOXES = 30

Class Box
    Private _width, _length, _height As Double

    'Define the constructor
    Public Sub New(width As Double, length As Double, height As Double)
        'Initialize fields
        Me._width = width
        Me._length = length
        Me._height = height
    End Sub

    Public Sub display_volume()
        Console.WriteLine("Volume: " & Me._width * Me._length * Me._height)
    End Sub

    Public Sub display_dimensions()
        Console.WriteLine(Me._width & " x " & Me._length & " x " & Me._height)
    End Sub
End Class

Sub Main(args As String())
    Dim i As Integer
    Dim w, l, h As Double

    Dim array_of_obj(BOXES) As Box 'Create an array

    For i = 0 To BOXES - 1
        Console.Write("Enter width: ")
        w = Console.ReadLine()
        Console.Write("Enter length: ")
        l = Console.ReadLine()
        Console.Write("Enter height: ")
        h = Console.ReadLine()

        'Add each new object to the array
        array_of_obj(i) = New Box(w, l, h)
    Next

    For i = 0 To BOXES - 1
        array_of_obj(i).display_dimensions()
        array_of_obj(i).display_volume()
    Next
```

```
End Sub
```

5. Solution

```
Const BOXES = 30

Class Box
    Private _width, _length, _height As Double

    'Define the constructor
    Public Sub New(width As Double, length As Double, height As Double)
        'Initialize properties
        Me.Width = width
        Me.Length = length
        Me.Height = height
    End Sub

    'Define public property Width
    Public Property Width As Double
        'Define the getter
        Get
            Return Me._width
        End Get
        'Define the setter
        Set
            If Value > 0 Then
                Me._width = Value
            Else
                Throw New Exception("Cannot be negative or zero")
            End If
        End Set
    End Property

    'Define public property Length
    Public Property Length As Double
        'Define the getter
        Get
            Return Me._length
        End Get
        'Define the setter
        Set
            If Value > 0 Then
                Me._length = Value
            Else
                Throw New Exception("Cannot be negative or zero")
            End If
        End Set
```

```
    End Property

    'Define public property Height
    Public Property Height As Double
        'Define the getter
        Get
            Return Me._height
        End Get
        'Define the setter
        Set
            If Value > 0 Then
                Me._height = Value
            Else
                Throw New Exception("Cannot be negative or zero")
            End If
        End Set
    End Property

    Public Sub display_volume()
        Console.WriteLine("Volume: " & Me.Width * Me.Length * Me.Height)
    End Sub

    Public Sub display_dimensions()
        Console.WriteLine(Me.Width & " x " & Me.Length & " x " & Me.Height)
    End Sub
End Class

Sub Main(args As String())
    Dim i As Integer
    Dim w, l, h As Double

    Dim array_of_obj(BOXES) As Box 'Create an array

    For i = 0 To BOXES - 1
        Console.Write("Enter width: ")
        w = Console.ReadLine()
        Console.Write("Enter length: ")
        l = Console.ReadLine()
        Console.Write("Enter height: ")
        h = Console.ReadLine()

        'Add each new object to the array
        array_of_obj(i) = New Box(w, l, h)
    Next

    For i = 0 To BOXES - 1
        array_of_obj(i).display_dimensions()
```

```
        array_of_obj(i).display_volume()
    Next
End Sub
```

6. Solution

```
Class Cube
    Private _edge As Double

    'Define the constructor
    Public Sub New(edge As Double)
        Me._edge = edge   'Initialize field
    End Sub

    Public Sub display_volume()
        Console.WriteLine("Volume: " & Me._edge ^ 3)
    End Sub

    Public Sub display_one_surface()
        Console.WriteLine("One surface: " & Me._edge ^ 2)
    End Sub

    Public Sub display_total_surface()
        Console.WriteLine("Total surface: " & 6 * Me._edge ^ 2)
    End Sub
End Class

Sub Main(args As String())

    Dim edge As Double

    Console.Write("Enter edge length of a cube: ")
    edge = Console.ReadLine()

    Dim cube1 As New Cube(edge)

    cube1.display_volume()
    cube1.display_one_surface()
    cube1.display_total_surface()
End Sub
```

7. Solution

```
Class Cube
    Private _edge As Double

    'Define the constructor
    Public Sub New(edge As Double)
        Me.Edge = edge   'Initialize property
    End Sub
```

```
    'Define public property Edge
    Public Property Edge As Double
        'Define the getter
        Get
            Return Me._edge
        End Get
        'Define the setter
        Set
            If value > 0 Then
                Me._edge = value
            Else
                Throw New Exception("Cannot be negative or zero")
            End If
        End Set
    End Property

    Public Sub display_volume()
        Console.WriteLine("Volume: " & Me.Edge ^ 3)
    End Sub

    Public Sub display_one_surface()
        Console.WriteLine("One surface: " & Me.Edge ^ 2)
    End Sub

    Public Sub display_total_surface()
        Console.WriteLine("Total surface: " & (6 * Me.Edge ^ 2))
    End Sub
End Class

Sub Main(args As String())
    Dim edge As Double

    Console.Write("Enter edge length of a cube: ")
    edge = Console.ReadLine()

    Dim cube1 As New Cube(edge)

    cube1.display_volume()
    cube1.display_one_surface()
    cube1.display_total_surface()
End Sub
```

8. Solution

```
Sub display_menu()
    Console.WriteLine("1. Enter radius")
    Console.WriteLine("2. Display radius")
```

```
    Console.WriteLine("3. Display diameter")
    Console.WriteLine("4. Display area")
    Console.WriteLine("5. Display perimeter")
    Console.WriteLine("6. Exit")
End Sub

Class Circle
    Private _radius As Double = -1

    'Define public property Radius
    Public Property Radius As Double
        'Define the getter
        Get
            If Me._radius > 0 Then
                Return Me._radius
            Else
                Throw New Exception("Radius is not set")
            End If
        End Get
        'Define the setter
        Set
            If Value > 0 Then
                Me._radius = Value
            Else
                Throw New Exception("Cannot be negative or zero")
            End If
        End Set
    End Property

    Public Function get_diameter() As Double
        Return 2 * Me.Radius
    End Function

    Public Function get_area() As Double
        Return 3.14 * Me.Radius ^ 2
    End Function

    Public Function get_perimeter() As Double
        Return 2 * 3.14 * Me.Radius
    End Function
End Class

Sub Main(args As String())
    Dim choice As Integer
    Dim radius As Double

    Dim circle1 As New Circle()
```

```
    display_menu()
    Console.Write("Enter a choice: ")
    choice = Console.ReadLine()
    Do While choice <> 6
        If choice = 1 Then
            Console.Write("Enter radius: ")
            radius = Console.ReadLine()
            circle1.Radius = radius
        ElseIf choice = 2 Then
            Console.WriteLine("Radius: " & circle1.Radius)
        ElseIf choice = 3 Then
            Console.WriteLine("Diameter: " & circle1.get_diameter())
        ElseIf choice = 4 Then
            Console.WriteLine("Area: " & circle1.get_area())
        ElseIf choice = 5 Then
            Console.WriteLine("Perimeter: " & circle1.get_perimeter())
        End If

        display_menu()
        Console.Write("Enter a choice: ")
        choice = Console.ReadLine()
    Loop
End Sub
```

9. Solution

```
Class Info
    Private _user_text As String

    'Define public property User_text
    Public Property User_text As String
        'Define the getter
        Get
            Return Me._user_text
        End Get
        'Define the setter
        Set
            If Value <> "" Then
                Me._user_text = Value
            Else
                Throw New Exception("Cannot be set to empty")
            End If
        End Set
    End Property

    Public Function get_spaces_count() As Integer
```

```
        Dim count As Integer = 0
        For Each character In Me.User_text
            If character.ToString() = " " Then
                count += 1
            End If
        Next
        Return count
    End Function

    Public Function get_words_count() As Integer
        Return Me.get_spaces_count() + 1
    End Function

    Public Function get_vowels_count() As Integer
        Dim count As Integer = 0
        For Each character In Me.User_text.ToLower()
            If "aeiou".IndexOf(character) > -1 Then
                count += 1
            End If
        Next
        Return count
    End Function

    Public Function get_letters_count() As Double
        Return Me.User_text.Length - Me.get_spaces_count()
    End Function
End Class

Sub Main(args As String())
    Dim inf As New Info()

    Console.Write("Enter a text: ")
    inf.User_text = Console.ReadLine()

    Console.WriteLine("Text: " & inf.User_text)
    Console.WriteLine("Spaces: " & inf.get_spaces_count())
    Console.WriteLine("Words: " & inf.get_words_count())
    Console.WriteLine("Vowels: " & inf.get_vowels_count())
    Console.WriteLine("Total number of letters: " & inf.get_letters_count())
End Sub
```

10.Solution

```
Sub display_menu()
    Console.WriteLine("1. Encryption/Decryption key")
    Console.WriteLine("2. Encrypt a message")
    Console.WriteLine("3. Decrypt a message")
    Console.WriteLine("4. Exit")
```

```
End Sub

Class EncryptDecrypt
    Private Const alphabet = " abcdefghijklmnopqrstuvwxyz"  'Space is a valid
    'character!
    Private _encr_decr_key As Integer = -1

    'Define public property Encr_decr_key
    Public Property Encr_decr_key As Integer
        'Define the getter
        Get
            If _encr_decr_key <> -1 Then
                Return Me._encr_decr_key
            Else
                Throw New Exception("Key is not set")
            End If
        End Get
        'Define the setter
        Set
            If value >= 1 And value <= 26 Then
                Me._encr_decr_key = value
            Else
                Throw New Exception("Must be between 1 and 26")
            End If
        End Set
    End Property

    Public Function encrypt(message As String) As String
        Dim new_letter As String, return_value As String = ""
        Dim index, new_index As Integer

        For Each character In message
            index = alphabet.IndexOf(character)
            new_index = index + Me.Encr_decr_key
            If new_index >= 27 Then
                new_index -= 27
            End If
            new_letter = alphabet(new_index).ToString()
            return_value += new_letter
        Next
        Return return_value
    End Function

    Public Function decrypt(enc_message As String) As String
        Dim new_letter As String, return_value As String = ""
        Dim index, new_index As Integer
```

```
        For Each character In enc_message
            index = alphabet.IndexOf(character)
            new_index = index - Me.Encr_decr_key
            If new_index < 0 Then
                new_index += 27
            End If
            new_letter = alphabet(new_index).ToString()
            return_value += new_letter
        Next
        Return return_value
    End Function
End Class

Sub Main(args As String())
    Dim text As String
    Dim choice As Integer

    Dim ed As New EncryptDecrypt()

    display_menu()
    Console.Write("Enter a choice: ")
    choice = Console.ReadLine()
    Do While choice <> 4
        If choice = 1 Then
            Console.Write("Enter encryption/decryption key: ")
            ed.Encr_decr_key = Console.ReadLine()
        ElseIf choice = 2 Then
            Console.Write("Enter message to encrypt: ")
            text = Console.ReadLine()
            Console.WriteLine("Encrypted message: " & ed.encrypt(text))
        ElseIf choice = 3 Then
            Console.Write("Enter message to decrypt: ")
            text = Console.ReadLine()
            Console.WriteLine("Decrypted message: " & ed.decrypt(text))
        End If

        display_menu()
        Console.Write("Enter a choice: ")
        choice = Console.ReadLine()
    Loop
End Sub
```

11.Solution

```
Class Vehicle
    Public number_of_wheels As Integer
    Public color As String
```

```
    Public length, width, height As Double

    'Define the constructor
    Public Sub New(number_of_wheels As Integer, color As String, length As Double,
                   width As Double, height As Double)
        Me.number_of_wheels = number_of_wheels
        Me.color = color
        Me.length = length
        Me.width = width
        Me.height = height
    End Sub

    Public Sub start_engine()
        Console.WriteLine("The engine started")
    End Sub

    Public Sub stop_engine()
        Console.WriteLine("The engine stopped")
    End Sub
End Class

Class Car
    Inherits Vehicle

    Public boot_capacity As Integer

    'Define the constructor
    Public Sub New(number_of_wheels As Integer, color As String, length As Double,
                   width As Double, height As Double)
        MyBase.New(number_of_wheels, color, length, width, height)
        Me.boot_capacity = 0
    End Sub

    Public Sub turn_windshield_wipers_on()
        Console.WriteLine("The windshield wipers have been turned on!")
    End Sub
End Class

Class Motorcycle
    Inherits Vehicle

    Public has_luggage As Boolean

    'Define the constructor
    Public Sub New(number_of_wheels As Integer, color As String, length As Double,
                   width As Double, height As Double)
        MyBase.New(number_of_wheels, color, length, width, height)
```

```
        Me.has_luggage = False
    End Sub

    Public Sub do_a_wheelie()
        Console.WriteLine("I am doing a wheelie!!!")
    End Sub
End Class

Sub Main(args As String())
    Dim car1 As New Car(4, "Red", 5, 2, 1.5)
    car1.boot_capacity = 300
    car1.start_engine()
    car1.turn_windshield_wipers_on()
    car1.stop_engine()

    Dim car2 As New Car(4, "Green", 4.5, 2.2, 1.4)
    car2.boot_capacity = 400
    car2.start_engine()
    car2.turn_windshield_wipers_on()
    car2.stop_engine()

    Dim motorcycle1 As New Motorcycle(2, "Blue", 2, 0.9, 1.3)
    motorcycle1.has_luggage = True
    motorcycle1.start_engine()
    motorcycle1.do_a_wheelie()
    motorcycle1.stop_engine()
End Sub
```

12.Solution

```
Class SchoolMember
    Private _name As String
    Private _age As Integer

    'Define the constructor
    Public Sub New(name As String, age As Integer)
        Me.setName(name)
        Me.setAge(age)
        Console.WriteLine("A school member was initialized")
    End Sub

    'Define the getter
    Public Function getName() As String
        Return Me._name
    End Function

    'Define the setter
    Public Sub setName(value As String)
```

```
        If value <> "" Then
            Me._name = value
        Else
            Throw New Exception("Name cannot be empty")
        End If
    End Sub

    'Define the getter
    Public Function getAge() As Integer
        Return Me._age
    End Function

    'Define the setter
    Public Sub setAge(value As Integer)
        If value > 0 Then
            Me._age = value
        Else
            Throw New Exception("Age cannot be negative or zero")
        End If
    End Sub
End Class

Class Teacher
    Inherits SchoolMember
    Private _salary As Double  'This is an additional field for this class

    'Define the constructor
    Public Sub New(name As String, age As Integer, salary As Double)
        MyBase.New(name, age)
        Me.setSalary(salary)

        'This is an additional statement for this constructor
        Console.WriteLine("A teacher was initialized")
    End Sub

    'Define the getter
    Public Function getSalary() As Double
        Return Me._salary
    End Function

    'Define the setter
    Public Sub setSalary(value As Double)
        If value >= 0 Then
            Me._salary = value
        Else
            Throw New Exception("Salary cannot be negative")
        End If
```

```
    End Sub

    'This is an additional method for this class
    Public Sub display_values()
        Console.WriteLine("Name: " & Me.getName())
        Console.WriteLine("Age: " & Me.getAge())
        Console.WriteLine("Salary: " & Me.getSalary())
    End Sub
End Class

Class Student
    Inherits SchoolMember
    Private _final_grade As String  'This is an additional field for this class

    'Define the constructor
    Public Sub New(name As String, age As Integer, final_grade As String)
        MyBase.New(name, age)
        Me.setFinalGrade(final_grade)

        'This is an additional statement for this constructor
        Console.WriteLine("A student was initialized")
    End Sub

    'Define the getter
    Public Function getFinalGrade() As String
        Return Me._final_grade
    End Function

    'Define the setter
    Public Sub setFinalGrade(value As String)
        If value <> "" Then
            Me._final_grade = value
        Else
            Throw New Exception("Final grade cannot be empty")
        End If
    End Sub

    'This is an additional method for this class
    Public Sub display_values()
        Console.WriteLine("Name: " & Me.getName())
        Console.WriteLine("Age: " & Me.getAge())
        Console.WriteLine("Final grade: " & Me.getFinalGrade())
    End Sub
End Class

Sub Main(args As String())
    Dim teacher1 As New Teacher("Mr. John Scott", 43, 35000)
```

```
    Dim teacher2 As New Teacher("Mrs. Ann Carter", 5, 32000)

    Dim student1 As New Student("Mark Nelson", 14, "A")
    Dim student2 As New Student("Mary Morgan", 13, "B")

    teacher1.display_values()
    teacher2.display_values()
    student1.display_values()
    student2.display_values()
End Sub
```

13.Solution

```
Class SchoolMember
    Private _name As String
    Private _age As Integer

    'Define the constructor
    Public Sub New(name As String, age As Integer)
        Me.Name = name
        Me.Age = age
        Console.WriteLine("A school member was initialized")
    End Sub

    'Define public property Name
    Public Property Name As String
        'Define the getter
        Get
            Return Me._name
        End Get

        'Define the setter
        Set
            If Value <> "" Then
                Me._name = Value
            Else
                Throw New Exception("Name cannot be empty")
            End If
        End Set
    End Property

    'Define public property Age
    Public Property Age As Integer
        'Define the getter
        Get
            Return Me._age
        End Get
```

```
        'Define the setter
        Set
            If Value > 0 Then
                Me._age = Value
            Else
                Throw New Exception("Age cannot be negative or zero")
            End If
        End Set
    End Property
End Class

Class Teacher
    Inherits SchoolMember
    Private _salary As Double  'This is an additional field for class Teacher

    'Define the constructor
    Public Sub New(name As String, age As Integer, salary As Double)
        MyBase.New(name, age)
        Me.Salary = salary
        Console.WriteLine("A teacher was initialized")
    End Sub

    'Define public property Salary
    Public Property Salary As Double
        'Define the getter
        Get
            Return Me._salary
        End Get

        'Define the setter
        Set
            If Value >= 0 Then
                Me._salary = Value
            Else
                Throw New Exception("Salary cannot be negative")
            End If
        End Set
    End Property

    'This is an additional method for this class
    Public Sub display_values()
        Console.WriteLine("Name: " & Me.Name)
        Console.WriteLine("Age: " & Me.Age)
        Console.WriteLine("Salary: " & Me.Salary)
    End Sub
```

```
End Class

Class Student
    Inherits SchoolMember
    Private _final_grade As String  'This is an additional field for class Student

    'Define the constructor
    Public Sub New(name As String, age As Integer, final_grade As String)
        MyBase.New(name, age)
        Me.FinalGrade = final_grade
        Console.WriteLine("A student was initialized")
    End Sub

    'Define public property Grades
    Public Property FinalGrade As String
        'Define the getter
        Get
            Return Me._final_grade
        End Get

        'Define the setter
        Set
            If Value <> "" Then
                Me._final_grade = Value
            Else
                Throw New Exception("Final grade cannot be empty")
            End If
        End Set
    End Property

    'This is an additional method for this class
    Public Sub display_values()
        Console.WriteLine("Name: " & Me.Name)
        Console.WriteLine("Age: " & Me.Age)
        Console.WriteLine("Final Grade: " & Me.FinalGrade)
    End Sub
End Class

Sub Main(args As String())
    Dim teacher1 As New Teacher("Mr. John Scott", 43, 35000)
    Dim teacher2 As New Teacher("Mrs. Ann Carter", 5, 32000)

    Dim student1 As New Student("Mark Nelson", 14, "A")
    Dim student2 As New Student("Mary Morgan", 13, "B")

    teacher1.display_values()
    teacher2.display_values()
```

```
    student1.display_values()
    student2.display_values()
End Sub
```

Chapter 29

29.7 Review Questions: True/False

1. False
2. False
3. True
4. False
5. False
6. False
7. False
8. False
9. True
10. False
11. True
12. True
13. False
14. False
15. True
16. True
17. False
18. False
19. True
20. True
21. True

29.8 Review Exercises

1. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim days() As String = {"Sunday", "Monday", "Tuesday", "Wednesday",
                                "Thursday", "Friday", "Saturday"}

        Dim f As StreamWriter = File.CreateText(PATH & "days_of_week.txt")
        For Each d In days
            f.WriteLine(d)
        Next
        f.Close()
    End Sub
End Module
```

2. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim i As Integer
```

```
        Dim days(6) As String

        Dim f As StreamReader = File.OpenText(PATH & "days_of_week.txt")
        For i = 0 To 6
            days(i) = f.ReadLine()
        Next
        f.Close()

        For i = 6 To 0 Step -1
            Console.WriteLine(days(i))
        Next
    End Sub
End Module
```

3. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim f As StreamWriter = File.AppendText(PATH & "days_of_week.txt")
        f.WriteLine("*** End of File ***")
        f.Close()
    End Sub
End Module
```

4. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim i As Integer
        Dim rnd As New Random()

        Dim f As StreamWriter = File.CreateText(PATH & "randoms.txt")
        For i = 0 To 49
            f.WriteLine(rnd.Next(1, 101))
        Next
        f.Close()
    End Sub
End Module
```

5. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim i As Integer
        Dim f As StreamWriter
        Dim rnd As New Random()

        For i = 1 To 10
            f = File.CreateText(PATH & "file" & i & ".txt")
            f.Write(rnd.Next(100, 10000))
            f.Close()
        Next
    End Sub
End Module
```

6. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim i, j As Integer
        Dim f As StreamWriter = File.CreateText(PATH & "multiplication_table.txt")

        For i = 1 To 10
            For j = 1 To 4
                f.WriteLine(i & " x " & j & " = " & i * j)
            Next
        Next
        f.Close()
    End Sub
End Module
```

7. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"
```

```
    Sub Main(args As String())
        Dim f As StreamReader = File.OpenText(PATH & "a_file.txt")

        Do While Not f.EndOfStream
            Console.WriteLine(f.ReadLine().Length)
        Loop
        f.Close()
    End Sub
End Module
```

8. Solution

First approach

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim i As Integer
        Dim line As String
        Dim f As StreamReader = File.OpenText(PATH & "a_file.txt")

        i = 1
        Do While Not f.EndOfStream
            line = f.ReadLine()
            For Each character In line
                If ",.!".IndexOf(character) > -1 Then
                    Console.WriteLine("There is a punctuation mark on line No " & i)
                    Exit For
                End If
            Next
            i += 1
        Loop

        f.Close()
    End Sub
End Module
```

Second approach

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim i As Integer
```

```
        Dim line As String
        Dim f As StreamReader = File.OpenText(PATH & "a_file.txt")

        i = 1
        Do While Not f.EndOfStream
            line = f.ReadLine()
            If line.IndexOf(",") > -1 Or line.IndexOf(".") > -1 Or line.IndexOf("!") > -1 Then
                Console.WriteLine("There is a punctuation mark on line No " & i)
            End If
            i += 1
        Loop

        f.Close()
    End Sub
End Module
```

Chapter 30

30.2 Review Exercises

1. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim total, count, number As Integer

        Dim fin As StreamReader = File.OpenText(PATH & "f_data30.2-1.txt")
        Dim values() As String = fin.ReadLine().Split()
        fin.Close()

        total = 0
        count = 0
        For Each value In values
            number = Convert.ToInt32(value)
            If number > 50 Then
                total += number
                count += 1
            End If
        Next

        If count > 0 Then
            Console.WriteLine(total / count)
```

```
        End If
    End Sub
End Module
```

2. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim total, count, number As Integer

        Dim fin As StreamReader = File.OpenText(PATH & "f_data30.2-2.txt")
        Dim values() As String = fin.ReadLine().Split(",")
        fin.Close()

        total = 0
        count = 0
        For Each value In values
            number = Convert.ToInt32(value)
            If number >= 300 And number <= 500 Then
                total += number
                count += 1
            End If
        Next

        If count > 0 Then
            Console.WriteLine(total / count)
        End If
    End Sub
End Module
```

3. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim max_name, min_name As String
        Dim maximum, minimum, grade As Integer

        Dim fin As StreamReader = File.OpenText(PATH & "f_data30.2-3.txt")

        'Read the first line
```

```
        Dim b() As String = fin.ReadLine().Split(",")

        maximum = Convert.ToInt32(b(0))
        minimum = maximum
        max_name = b(1)
        min_name = b(1)

        'Read the rest of the lines
        Do While Not fin.EndOfStream
            b = fin.ReadLine().Split(",")
            grade = Convert.ToInt32(b(0))

            If grade > maximum Then
                maximum = grade
                max_name = b(1)
            End If
            If grade < minimum Then
                minimum = grade
                min_name = b(1)
            End If
        Loop

        fin.Close()

        Console.WriteLine(max_name)
        Console.WriteLine(min_name)
    End Sub
End Module
```

4. Solution

First approach

```
Imports System
Imports System.IO

Module Program
    Sub Main(args As String())
        Dim filename1, filename2, content As String
        Dim fin As StreamReader
        Dim fout As StreamWriter

        Console.Write("Enter filename No 1: ")
        filename1 = Console.ReadLine()

        If filename1.Substring(filename1.Length - 4) <> ".txt" Then
            Console.WriteLine("Wrong filename")
        Else
            Console.Write("Enter filename No 2: ")
```

```
            filename2 = Console.ReadLine()
            If filename2.Substring(filename1.Length - 4) <> ".txt" Then
                Console.WriteLine("Wrong filename")
            Else
                fin = File.OpenText(filename2)
                content = fin.ReadToEnd()
                fin.Close()

                fin = File.OpenText(filename1)
                content &= fin.ReadToEnd()    'Concatenation
                fin.Close()

                fout = File.CreateText("final.txt")
                fout.Write(content)
                fout.Close()
            End If
        End If
    End Sub
End Module
```

Second approach

```
Imports System
Imports System.IO

Module Program
    Sub Main(args As String())
        Dim filename1, filename2 As String

        Console.Write("Enter filename No 1: ")
        filename1 = Console.ReadLine()

        If filename1.Substring(filename1.Length - 4) <> ".txt" Then
            Console.WriteLine("Wrong filename")
        Else
            Console.Write("Enter filename No 2: ")
            filename2 = Console.ReadLine()
            If filename2.Substring(filename1.Length - 4) <> ".txt" Then
                Console.WriteLine("Wrong filename")
            Else
                Dim fin1 As StreamReader = File.OpenText(filename1)
                Dim fin2 As StreamReader = File.OpenText(filename2)
                Dim fout As StreamWriter = File.CreateText("final.txt")

                fout.Write(fin2.ReadToEnd() & fin1.ReadToEnd())

                fout.Close()
                fin2.Close()
                fin1.Close()
```

```
            End If
        End If
    End Sub
End Module
```

5. Solution

First approach – For 15 numbers

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"
    Const ELEMENTS = 15

    Sub Main(args As String())
        Dim i As Integer
        Dim numbers(ELEMENTS - 1) As Double

        Dim fin As StreamReader = File.OpenText(PATH & "f_data30.2-5.txt")
        For i = 0 To ELEMENTS - 1
            numbers(i) = Convert.ToDouble(fin.ReadLine())
        Next
        fin.Close()

        Array.Sort(numbers)

        Dim fout As StreamWriter = File.AppendText(PATH & "f_data30.2-5.txt")
        fout.WriteLine(vbCrLf & "********* Sorted numbers *************")
        For Each number In numbers
            fout.WriteLine(number)
        Next
        fout.Close()
    End Sub
End Module
```

Second approach – For any number of numbers

```
Imports System
Imports System.IO

Module Program
Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim i As Integer

        'Create array s_numbers of type string
        Dim fin As StreamReader = File.OpenText(PATH & "f_data30.2-5.txt")
        Dim s_numbers() As String = fin.ReadToEnd().Split(vbCrLf)
```

```
        fin.Close()

        'Create array numbers of type double
        Dim numbers(s_numbers.Length - 1) As Double
        For i = 0 To s_numbers.Length - 1
            numbers(i) = Convert.ToDouble(s_numbers(i))
        Next

        Array.Sort(numbers)

        Dim fout As StreamWriter = File.AppendText(PATH & "f_data30.2-5.txt")
        fout.WriteLine(vbCrLf & "********* Sorted numbers *************")
        For Each number In numbers
            fout.WriteLine(number)
        Next
        fout.Close()
    End Sub
End Module
```

6. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"

    Sub Main(args As String())
        Dim i, j As Integer
        Dim onCityLine As Boolean
        Dim total, average, maximum As Double

        Dim fin As StreamReader = File.OpenText(PATH & "f_data30.2-6.txt")
        Dim s_values() As String = fin.ReadToEnd().Split(vbCrLf)
        fin.Close()

        Dim cities(s_values.Length / 2 - 1) As String
        Dim temperatures(s_values.Length / 2 - 1) As Double

        'Split array s_values into two arrays (cities and temperatures)
        i = 0
        j = 0
        onCityLine = True
        For Each s_value In s_values
            If onCityLine Then
                cities(i) = s_value
                i += 1
            Else
```

```
                temperatures(j) = Convert.ToDouble(s_value)
                j += 1
            End If
            onCityLine = Not onCityLine  'True becomes False, and False becomes True
        Next

        total = 0
        For i = 0 To temperatures.Length - 1
            total += temperatures(i)
        Next

        average = total / temperatures.Length
        Console.WriteLine(average)

        maximum = temperatures.Max()
        Console.WriteLine("Highest temperature: " & maximum)
        For i = 0 To temperatures.Length - 1
            If temperatures(i) = maximum Then
                Console.WriteLine(cities(i))
            End If
        Next
    End Sub
End Module
```

7. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"
    Const x = " ABCDEFGHIJKLMNOPQRSTUVWXYZ"   'The space character remains as is
    Const y = " JKWCTAMEDXSLFBYUNGRZOIQVHP"

    Sub Main(args As String())
        Dim initial_message, encrypted_message As String
        Dim i As Integer

        Console.Write("Enter a message to encrypt: ")
        initial_message = Console.ReadLine().ToUpper()

        encrypted_message = ""
        For Each letter In initial_message
            'Search for letter in const x
            For i = 0 To x.Length - 1
                If letter = x(i) Then
                    'Create encrypted message using letters from const y
                    encrypted_message &= y(i)
```

```
                    Exit For
                End If
            Next
        Next

        Dim fout As StreamWriter = File.CreateText(PATH & "encrypted.txt")
        fout.Write(encrypted_message)
        fout.Close()
    End Sub
End Module
```

8. Solution

```
Imports System
Imports System.IO

Module Program
    Const PATH = "c:/temp/"
    Const x = " ABCDEFGHIJKLMNOPQRSTUVWXYZ"   'The space character remains as is
    Const y = " JKWCTAMEDXSLFBYUNGRZOIQVHP"

    Sub Main(args As String())
        Dim initial_message, encrypted_message As String
        Dim i As Integer

        Dim fin As StreamReader = File.OpenText(PATH & "encrypted.txt")
        encrypted_message = fin.ReadLine()
        fin.Close()

        initial_message = ""
        For Each letter In encrypted_message
            'Search for letter in const y
            For i = 0 To y.Length - 1
                If letter = y(i) Then
                    'Create decrypted message using letters from const x
                    initial_message &= x(i)
                    Exit For
                End If
            Next
        Next

        Dim fout As StreamWriter = File.CreateText(PATH & "decrypted.txt")
        fout.Write(initial_message)
        fout.Close()
    End Sub
End Module
```

9. Solution

First approach

```
Sub copy(source As String, destination As String)
    Dim fin As StreamReader = File.OpenText(source)
    Dim x As String = fin.ReadToEnd()
    fin.Close()

    Dim fout As StreamWriter = File.CreateText(destination)
    fout.Write(x)
    fout.Close()
End Sub
```

Second approach

```
Sub copy(source As String, destination As String)
    Dim fin As StreamReader = File.OpenText(source)
    Dim fout As StreamWriter = File.CreateText(destination)

    fout.Write(fin.ReadToEnd())

    fin.Close()
    fout.Close()
End Sub
```

10.Solution

```
Imports System
Imports System.IO

Module Program
    Class Triangle
        Const PATH = "c:/temp/"

        Private _sideA, _sideB, _sideC As Double

        'Define the constructor
        Public Sub New()
            Dim fin As StreamReader = File.OpenText(PATH & "f_data30.2-10.txt")
            Me._sideA = Convert.ToDouble(fin.ReadLine())
            Me._sideB = Convert.ToDouble(fin.ReadLine())
            Me._sideC = Convert.ToDouble(fin.ReadLine())
            fin.Close()
        End Sub

        Public Function can_be_triangle() As Boolean
            If Me._sideA > 0 And Me._sideB > 0 And Me._sideC > 0 And
               Me._sideA + Me._sideB > Me._sideC And
               Me._sideB + Me._sideC > Me._sideA And
               Me._sideC + Me._sideA > Me._sideB Then
               Return True
```

```
            Else
                Return False
            End If
        End Function

        Public Sub display_lengths()
            Console.WriteLine("Side A: " & Me._sideA)
            Console.WriteLine("Side B: " & Me._sideB)
            Console.WriteLine("Side C: " & Me._sideC)
            If Me.can_be_triangle() Then
                Console.WriteLine("Can be lengths of the three sides of a triangle!")
            Else
                Console.WriteLine("Cannot be lengths of the three sides of a triangle!")
            End If
        End Sub

        Public Sub display_area()
            Dim s, area As Double

            If Me.can_be_triangle() Then
                s = (Me._sideA + Me._sideB + Me._sideC) / 2
                area = Math.Sqrt(s * (s - Me._sideA) * (s - Me._sideB) * (s - Me._sideC))
                Console.WriteLine("Area: " & area)
            End If
        End Sub

        Public Sub display_perimeter()
            Dim perimeter As Double

            If Me.can_be_triangle() Then
                perimeter = Me._sideA + Me._sideB + Me._sideC
                Console.WriteLine("Perimeter: " & perimeter)
            End If
        End Sub
    End Class

    Sub Main(args As String())
        Dim tr As New Triangle()

        tr.display_lengths()
        tr.display_area()
        tr.display_perimeter()
    End Sub
End Module
```