C# for Tweens and Teens

The Answers

By
Aristides S. Bouras
Loukia V. Ainarozidou

C# for Tweens and Teens - The Answers

http://www.bouraspage.com

RCode: 170718

Warning and Disclaimer

This book is designed to provide the answers to all of the review questions, as well as the solutions to all review exercises of the book "C# for Tweens and Teens". Every effort has been taken to make this book compatible with all releases of Python 3.x, and it is almost certain to be compatible with any future releases of Python.

The information is provided on an "as is" basis. The authors 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 6
Chapter 1 7
1.7 Review Questions: True/False 7
1.8 Review Questions: Multiple Choice 7
Chapter 3 7
3.11 Review Questions: True/False 7
3.12 Review Questions: Multiple Choice 7
Chapter 4 7
4.8 Review Questions: True/False 7
4.9 Review Questions: Multiple Choice 8
4.10 Review Exercises 8
Chapter 5 8
5.4 Review Questions: True/False 8
5.5 Review Questions: Multiple Choice 8
Chapter 6 8
6.7 Review Questions: True/False 8
6.8 Review Questions: Multiple Choice 8
6.9 Review Exercises 8
Chapter 8 9
8.2 Review Exercises 9
Chapter 9 11
9.4 Review Questions: True/False 11
9.5 Review Exercises 11
Chapter 10 12
10.4 Review Questions: True/False 12
10.5 Review Questions: Multiple Choice 12
10.6 Review Exercises 12
Chapter 11 13
11.7 Review Questions: True/False 13
11.8 Review Questions: Multiple Choice 13
11.9 Review Exercises 13
Chapter 12 15

12.2 Review Questions: True/False 15

12.3 Review Questions: Multiple Choice 15

12.4 Review Exercises 15

Chapter 13 20

13.2 Review Questions: True/False 20

13.3 Review Questions: Multiple Choice 20

13.4 Review Exercises 20

Chapter 14 22

14.2 Review Questions: True/False 22

14.3 Review Exercises 22

Chapter 15 31

15.2 Review Questions: True/False 31

15.3 Review Exercises 31

Chapter 16 33

16.3 Review Questions: True/False 33

Chapter 17 33

17.3 Review Questions: True/False 33

17.4 Review Questions: Multiple Choice 33

17.5 Review Exercises 33

Chapter 18 37

18.3 Review Questions: True/False 37

18.4 Review Questions: Multiple Choice 37

18.5 Review Exercises 37

Chapter 19 40

19.2 Review Questions: True/False 40

19.3 Review Questions: Multiple Choice 40

19.4 Review Exercises 40

Chapter 20 43

20.7 Review Questions: True/False 43

20.8 Review Questions: Multiple Choice 43

20.9 Review Exercises 43

Chapter 21 46

21.2 Review Exercises 46

Chapter 22 54

22.12 Review Exercises 54

Chapter 23 61

23.14 Review Questions: True/False 61

23.15 Review Questions: Multiple Choice 61

23.16 Review Exercises 62

Chapter 24 72

24.5 Review Questions: True/False 72

24.6 Review Exercises 72

Chapter 25 82

25.4 Review Questions: True/False 82

Chapter 26 83

26.11 Review Questions: True/False 83

26.12 Review Exercises 83

Chapter 27 85

27.2 Review Exercises 85

Chapter 28 90

28.8 Review Questions: True/False 90

28.9 Review Exercises 90

How to Report Errata

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

http://www.bouraspage.com

Once your errata are verified, your submission will be accepted and the errata will be uploaded to our 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. False
21. False
22. 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. c

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. False
14. False
15. True
16. 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

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. b
3. b

Chapter 6

6.7 Review Questions: True/False

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

6.8 Review Questions: Multiple Choice

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

6.9 Review Exercises

1. ii, iv, v, ix, x
2. i – String, ii – Boolean, iii – String, iv – String, v – Float, vi – Integer
3. i – c, ii – d, iii – c, iv – e

4. i – 27, ii – 28
5. i – 0, ii – 4
6. i – 2, 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

```
static void Main() {
    double b, h, area;

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

    area = b * h / 2;

    Console.WriteLine(area);
    Console.ReadKey();
}
```

2. Solution

```
static void Main() {
    double f, k;

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

    k = (f + 459.67) / 1.8;

    Console.WriteLine(k);
    Console.ReadKey();
}
```

3. Solution

```
static void Main() {
    double angle1, angle2, angle3;

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

    angle3 = 180 - angle1 - angle2;

    Console.WriteLine(angle3);
    Console.ReadKey();
}
```

4. Solution

```
static void Main() {
    double average;

    Console.Write("Enter 1st grade: ");
    g1 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter 2nd grade: ");
    g2 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter 3rd grade: ");
    g3 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter 4th grade: ");
    g4 = Int32.Parse(Console.ReadLine());

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

    Console.WriteLine(average);
    Console.ReadKey();
}
```

5. Solution

```
const double PI = 3.14159;
static void Main() {
    double r, perimeter;

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

    perimeter = 2 * PI * r;

    Console.WriteLine(perimeter);
    Console.ReadKey();
}
```

6. Solution

```
static void Main() {
    int w, h;
    double bmi;

    Console.Write("Enter weight in pounds: ");
    w = Int32.Parse(Console.ReadLine());
    Console.Write("Enter height in inches: ");
    h = Int32.Parse(Console.ReadLine());

    bmi = w * 703.0 / (h * h);

    Console.WriteLine(bmi);
    Console.ReadKey();
}
```

7. Solution

```
static void Main() {
    int day, month, days_passed, days_left;

    Console.Write("Enter day: ");
    day = Int32.Parse(Console.ReadLine());
    Console.Write("Enter month: ");
    month = Int32.Parse(Console.ReadLine());

    days_passed = (month - 1) * 30 + day;
    days_left = 360 - days_passed;

    Console.WriteLine(days_left);
    Console.ReadKey();
}
```

Chapter 9

9.4 Review Questions: True/False

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

9.5 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

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. False
10. True
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

```
static void Main() {
    string f_name, m_name, l_name, title;

    Console.Write("First name: ");
    f_name = Console.ReadLine();
    Console.Write("Middle name: ");
    m_name = Console.ReadLine();
    Console.Write("Last name: ");
    l_name = Console.ReadLine();
    Console.Write("Title: ");
    title = Console.ReadLine();

    Console.WriteLine(title + " " + f_name + " " + m_name + " " + l_name);
    Console.WriteLine(f_name + " " + m_name + " " + l_name);
    Console.WriteLine(l_name + ", " + f_name);
    Console.WriteLine(l_name + ", " + f_name + " " + m_name);
    Console.WriteLine(l_name + ", " + f_name + " " + m_name + ", " + title);
    Console.WriteLine(f_name + " " + l_name);
    Console.ReadKey();
}
```

2. Solution

```
static void Main() {
    string alphabet;
    Random rnd = new Random(Guid.NewGuid().GetHashCode());

    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)]);
    Console.ReadKey();
}
```

3. *Solution*

```
static void Main() {
    string name, x, secret_password;
    Random rnd = new Random(Guid.NewGuid().GetHashCode());

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

    //convert to lower case and remove spaces
    x = name.ToLower().Replace(" ", "");

    secret_password = x[rnd.Next(0, x.Length)].ToString() +
                      x[rnd.Next(0, x.Length)].ToString() +
                      x[rnd.Next(0, x.Length)].ToString() +
                      rnd.Next(1000, 10000).ToString();

    Console.WriteLine(secret_password);
    Console.ReadKey();
}
```

Chapter 11

11.7 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

11.8 Review Questions: Multiple Choice

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

11.9 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

Boolean Expression1 (BE1)	Boolean Expression2 (BE2)	BE1 || BE2	BE1 && BE2	!(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 || c > b && c > 1	a > 3 && c > b || c > 1
4	-6	2	*True*	*True*
-3	2	-4	*False*	*False*

5. Solution

Expression	Value
`Math.Pow(x + y, 3)`	*8.0*
`(x + y) / (Math.Pow(x, 2) - 14)`	*1.0*
`(x – 1) == y + 5`	*true*
`x > 2 && y == 1`	*false*
`x == 1 || !(flag == false)`	*true*

6. Solution

a. `age < 12 && age != 8`

b. `age >= 6 && age <= 9 || age == 11`

c. `age > 7 && age != 10 && age != 12`

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

e. `age >= 6 && age <= 12 && age != 8`

f. `age != 7 && 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

```
static void Main() {
  double x, y;

  x = Double.Parse(Console.ReadLine();

  y = - 5;
  if (x * y / 2 > 20) {
    y--;
    x += 4 * Math.Pow(x, 2);
  }
  Console.Write(x + " " + y);
  Console.ReadKey();
}
```

2. Solution

i. 9 12

ii. 2 2

3. Solution

```
static void Main() {
    double x;

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

    if (x > 0) {
        Console.WriteLine("Positive");
    }
    Console.ReadKey();
}
```

4. Solution

```
static void Main() {
    double x, y;

    Console.Write("Enter a number: ");
    x = Double.Parse(Console.ReadLine());
    Console.Write("Enter a second number");
    y = Double.Parse(Console.ReadLine());

    if (x > 0 && y > 0) {
        Console.WriteLine("Positives");
    }
    Console.ReadKey();
}
```

5. Solution

```
static void Main() {
    string s;

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

    if (s == s.ToUpper()) {
        Console.WriteLine("Uppercase");
    }
    Console.ReadKey();
}
```

6. Solution

```
static void Main() {
    string s;

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

    if (s.Length > 20) {
        Console.WriteLine("Many characters");
    }
    Console.ReadKey();
}
```

7. Solution

```
static void Main() {
    double n1, n2, n3;

    Console.Write("Enter 1st number: ");
    n1 = Double.Parse(Console.ReadLine());
    Console.Write("Enter 2nd number: ");
    n2 = Double.Parse(Console.ReadLine());
    Console.Write("Enter 3rd number: ");
    n3 = Double.Parse(Console.ReadLine());
    if (n1 < 0 || n2 < 0 || n3 < 0) {
        Console.WriteLine("Among the numbers, there is a negative one!");
    }
    Console.ReadKey();
}
```

8. Solution

```
static void Main() {
    double t1, t2, t3, average;

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

    average = (t1 + t2 + t3) / 3;

    if (average > 60) {
        Console.WriteLine("Heat Wave");
    }
    Console.ReadKey();
}
```

9. Solution

```
static void Main() {
    double w1, w2, w3, w4, maximum;

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

    maximum = w1;

    if (w2 > maximum) {
        maximum = w2;
    }

    if (w3 > maximum) {
        maximum = w3;
    }

    if (w4 > maximum) {
        maximum = w4;
    }

    Console.WriteLine(maximum);
    Console.ReadKey();
}
```

10. Solution

```
static void Main() {
    string n1, n2, n3, n4, m_name;
    int a1, a2, a3, a4, minimum;

    Console.Write("Enter the age of the 1st person: ");
    a1 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter the name of the 1st person: ");
    n1 = Console.ReadLine();

    Console.Write("Enter the age of the 2nd person: ");
    a2 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter the name of the 2nd person: ");
    n2 = Console.ReadLine();

    Console.Write("Enter the age of the 3rd person: ");
    a3 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter the name of the 3rd person: ");
    n3 = Console.ReadLine();

    Console.Write("Enter the age of the 4th person: ");
    a4 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter the name of the 4th person: ");
    n4 = Console.ReadLine();

    minimum = a1;
    m_name = n1;

    if (a2 < minimum) {
        minimum = a2;
        m_name = n2;
    }

    if (a3 < minimum) {
        minimum = a3;
        m_name = n3;
    }
    if (a4 < minimum) {
        minimum = a4;
        m_name = n4;
    }

    Console.WriteLine("The youngest person is " + m_name);
    Console.ReadKey();
}
```

11. Solution

```
static void Main() {
    int a1, a2, a3, minimum, maximum, middle;

    Console.Write("Enter the age of the 1st person: ");
    a1 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter the age of the 2nd person: ");
    a2 = Int32.Parse(Console.ReadLine());
    Console.Write("Enter the age of the 3rd person: ");
    a3 = Int32.Parse(Console.ReadLine());

    minimum = a1;
    if (a2 < minimum) {
        minimum = a2;
    }

    if (a3 < minimum) {
        minimum = a3;
    }

    maximum = a1;
    if (a2 > maximum) {
        maximum = a2;
    }

    if (a3 > maximum) {
        maximum = a3;
    }

    middle = a1 + a2 + a3 - minimum - maximum;

    Console.WriteLine(middle);
    Console.ReadKey();
}
```

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

```
static void Main() {
    double num;

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

    if (num > 100) {
        Console.WriteLine("Given number is greater than 100");
    }
    else {
        Console.WriteLine("Given number is less than or equal to 100");
    }
    Console.ReadKey();
}
```

4. Solution

```
static void Main() {
    double num;

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

    if (num >= 0 && num <= 100) {
        Console.WriteLine("Given number is between 0 and 100");
    }
    else {
        Console.WriteLine("Given number is not between 0 and 100");
    }
    Console.ReadKey();
}
```

5. Solution

```
static void Main() {
    int num;

    Console.Write("Enter an integer: ");
    num = Int32.Parse(Console.ReadLine());

    if (num >= 1000 && num <= 9999) {
        Console.WriteLine("Given number is a four-digit integer");
    }
    else {
        Console.WriteLine("Given number is not a four-digit integer");
    }
    Console.ReadKey();
}
```

6. Solution

```
static void Main() {
    double num1, num2;

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

    if (num1 < num2) {
        Console.WriteLine(num1);
    }
    else {
        Console.WriteLine(num2);
    }
    Console.ReadKey();
}
```

7. Solution

```
static void Main() {
    double a, b, c, average;

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

    average = (a + b + c) / 3;

    if (average >= 8) {
        Console.WriteLine("Qualified");
    }
    else {
        Console.WriteLine("Disqualified");
    }
    Console.ReadKey();
}
```

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 ii. 10 90 iii. 20 160

3. Solution

First Approach

```
static void Main() {
    int a, n;

    Console.Write("Enter an integer between -9999 and 9999: ");
    a = Int32.Parse(Console.ReadLine());

    if (a >= -9999 && a <= -1000 || a >= 1000 && a <= 9999) {
        n = 4;
    }
    else if (a >= -999 && a <= -100 || a >= 100 && a <= 999) {
        n = 3;
    }
    else if (a >= -99 && a <= -10 || a >= 10 && a <= 99) {
        n = 2;
    }
    else {
        n = 1;
    }

    Console.Write("You entered a " + n + "-digit integer");
    Console.ReadKey();
}
```

Second Approach

```
static void Main() {
    int a, n;

    Console.Write("Enter an integer between -9999 and 9999: ");
    a = Int32.Parse(Console.ReadLine());

    //If variable a is negative, make it positive
    if (a < 0) {
        a = (-1) * a;
    }

    if (a >= 1000 && a <= 9999) {
        n = 3;
    }
    else if (a >= 100 && a <= 999) {
        n = 3;
    }
    else if (a >= 10 && a <= 99) {
        n = 2;
    }
    else {
        n = 1;
    }

    Console.Write("You entered a " + n + "-digit integer");
    Console.ReadKey();
}
```

4. Solution

```
static void Main() {
    int m;

    Console.Write("Enter the number of a month between 1 and 12: ");
    m = Int32.Parse(Console.ReadLine());

    if (m <= 2 || m == 12) {
        Console.WriteLine("Winter");
    }
    else if (m <= 5) {
        Console.WriteLine("Spring");
    }
    else if (m <= 8) {
        Console.WriteLine("Summer");
    }
    else {
        Console.WriteLine("Fall (Autumn)");
    }
    Console.ReadKey();
}
```

5. Solution

```
static void Main() {
    string letter;

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

    if (letter == "A") {
        Console.WriteLine("90 - 100");
    }
    else if (letter == "B") {
        Console.WriteLine("80 - 89");
    }
    else if (letter == "C") {
        Console.WriteLine("70 - 79");
    }
    else if (letter == "D") {
        Console.WriteLine("60 - 69");
    }
    else {
        Console.WriteLine("0 - 59");
    }
    Console.ReadKey();
}
```

6. Solution

```
static void Main() {
    string name;

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

    if (name == "JANUARY") {
        Console.WriteLine(1);
    }
    else if (name == "FEBRUARY") {
        Console.WriteLine(2);
    }
    else if (name == "MARCH") {
        Console.WriteLine(3);
    }
    else if (name == "APRIL") {
        Console.WriteLine(4);
    }
    else if (name == "MAY") {
        Console.WriteLine(5);
    }
    else if (name == "JUNE") {
        Console.WriteLine(6);
    }
    else if (name == "JULY") {
        Console.WriteLine(7);
    }
    else if (name == "AUGUST") {
        Console.WriteLine(8);
    }
    else if (name == "SEPTEMBER") {
        Console.WriteLine(9);
    }
    else if (name == "OCTOBER") {
        Console.WriteLine(10);
    }
    else if (name == "NOVEMBER") {
        Console.WriteLine(11);
    }
    else if (name == "DECEMBER") {
        Console.WriteLine(12);
    }
    else {
        Console.WriteLine("Error");
    }
    Console.ReadKey();
}
```

7. Solution

```
static void Main() {
    string roman;

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

    if (roman == "I") {
        Console.WriteLine(1);
    }
    else if (roman == "II") {
        Console.WriteLine(2);
    }
    else if (roman == "III") {
        Console.WriteLine(3);
    }
    else if (roman == "IV") {
        Console.WriteLine(4);
    }
    else if (roman == "V") {
        Console.WriteLine(5);
    }
    else if (roman == "VI") {
        Console.WriteLine(6);
    }
    else if (roman == "VII") {
        Console.WriteLine(7);
    }
    else if (roman == "VIII") {
        Console.WriteLine(8);
    }
    else if (roman == "IX") {
        Console.WriteLine(9);
    }
    else if (roman == "X") {
        Console.WriteLine(10);
    }
    else {
        Console.WriteLine("Error");
    }
    Console.ReadKey();
}
```

8. Solution

```
static void Main() {
    int total ;

    Console.Write("Enter the total number of CDs purchased in a month: ");
    total = Int32.Parse(Console.ReadLine());

    if (total == 1) {
        Console.WriteLine("You are awarded 3 points");
    }
    else if (total == 2) {
        Console.WriteLine("You are awarded 10 points");
    }
    else if (total == 3) {
        Console.WriteLine("You are awarded 20 points");
    }
    else {
        Console.WriteLine("You are awarded 45 points");
    }
    Console.ReadKey();
}
```

9. Solution

```
static void Main() {
    string num;

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

    if (num == "zero") {
        Console.WriteLine(0);
    }
    else if (num == "one") {
        Console.WriteLine(1);
    }
    else if (num == "two") {
        Console.WriteLine(2);
    }
    else if (num == "three") {
        Console.WriteLine(3);
    }
    else {
        Console.WriteLine("I don't know this number!");
    }
    Console.ReadKey();
}
```

10. Solution

```
static void Main() {
    int b;

    Console.Write("Enter Beaufort number: ");
    b = Int32.Parse(Console.ReadLine());

    if (b == 0) {
        Console.WriteLine("Calm");
    }
    else if (b == 1) {
        Console.WriteLine("Light Air");
    }
    else if (b == 2) {
        Console.WriteLine("Light breeze");
    }
    else if (b == 3) {
        Console.WriteLine("Gentle breeze");
    }
    else if (b == 4) {
        Console.WriteLine("Moderate breeze");
    }
    else if (b == 5) {
        Console.WriteLine("Fresh breeze");
    }
    else if (b == 6) {
        Console.WriteLine("Strong breeze");
    }
    else if (b == 7) {
        Console.WriteLine("Moderate gale");
    }
    else if (b == 8) {
        Console.WriteLine("Gale");
    }
    else if (b == 9) {
        Console.WriteLine("Strong gale");
    }
    else if (b == 10) {
        Console.WriteLine("Storm");
    }
    else if (b == 11) {
        Console.WriteLine("Violent storm");
    }
    else if (b == 12) {
        Console.WriteLine("Hurricane force");
    }
    else {
        Console.WriteLine("Invalid Beaufort number!");
    }
    Console.ReadKey();
}
```

11. Solution

```
static void Main() {
    double wind;

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

    if (wind < 0) {
        Console.WriteLine("Entered value is negative");
    }
    else if (wind < 1) {
        Console.Write("Beaufort: 0\nCalm");
    }
    else if (wind < 4) {
        Console.Write("Beaufort: 1\nLight air");
    }
    else if (wind < 8) {
        Console.Write("Beaufort: 2\nLight breeze");
    }
    else if (wind < 13) {
        Console.Write("Beaufort: 3\nGentle breeze");
    }
    else if (wind < 18) {
        Console.Write("Beaufort: 4\nModerate breeze");
    }
    else if (wind < 25) {
        Console.Write("Beaufort: 5\nFresh breeze");
    }
    else if (wind < 31) {
        Console.Write("Beaufort: 6\nStrong breeze");
    }
    else if (wind < 39) {
        Console.Write("Beaufort: 7\nModerate gale");
    }
    else if (wind < 47) {
        Console.Write("Beaufort: 8\nGale");
    }
    else if (wind < 55) {
        Console.Write("Beaufort: 9\nStrong gale");
    }
    else if (wind < 64) {
        Console.Write("Beaufort: 10\nStorm");
    }
    else if (wind < 74) {
        Console.Write("Beaufort: 11\nViolent storm");
    }
    else {
        Console.Write("Beaufort: 12\nHurricane force");
    }
    Console.ReadKey();
}
```

12. Solution

```
static void Main() {
    int choice;
    double kelvin, fahrenheit, celsius;

    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 = Int32.Parse(Console.ReadLine());

    if (choice == 1) {
        Console.Write("Enter a temperature in degrees Kelvin: ");
        kelvin = Double.Parse(Console.ReadLine());
        fahrenheit = 1.8 * kelvin - 459.67;
        Console.Write(kelvin + " degrees Kelvin = ");
        Console.Write(fahrenheit + " degrees Fahrenheit");
    }
    else if (choice == 2) {
        Console.Write("Enter a temperature in degrees Fahrenheit: ");
        fahrenheit = Double.Parse(Console.ReadLine());
        kelvin = (fahrenheit + 459.67) / 1.8;
        Console.Write(fahrenheit + " degrees Fahrenheit = ");
        Console.Write(kelvin + " degrees Kelvin");
    }
    else if (choice == 3) {
        Console.Write("Enter a temperature in degrees Fahrenheit: ");
        fahrenheit = Double.Parse(Console.ReadLine());
        celsius = 5.0 / 9.0 * (fahrenheit - 32);
        Console.Write(fahrenheit + " degrees Fahrenheit = ");
        Console.Write(celsius + " degrees Celsius");
    }
    else if (choice == 4) {
        Console.Write("Enter a temperature in degrees Celsius: ");
        celsius = Double.Parse(Console.ReadLine());
        fahrenheit = 9.0 / 5.0 * celsius + 32;
        Console.Write(celsius + " degrees Celsius = ");
        Console.Write(fahrenheit + " degrees Fahrenheit");
    }
    else {
        Console.WriteLine("Invalid choice!");
    }
    Console.ReadKey();
}
```

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

```
static void Main() {
    double t, w;

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

    if (t > 75) {
        if (w > 12) {
            Console.WriteLine("The day is hot and windy");
        }
        else {
            Console.WriteLine("The day is hot and not windy");
        }
    }
    else {
        if (w > 12) {
            Console.WriteLine("The day is cold and windy");
        }
        else {
            Console.WriteLine("The day is cold and not windy");
        }
    }
    Console.ReadKey();
}
```

Second Approach

```
static void Main() {
    double t, w;
    string message1, message2;

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

    if (t > 75) {
        message1 = "hot";
    }
    else {
        message1 = "cold";
    }

    if (w > 12) {
        message2 = "windy";
    }
    else {
        message2 = "not windy";
    }

    Console.WriteLine("The day is " + message1 + " and " + message2);
    Console.ReadKey();
}
```

3. Solution

```
static void Main() {
    int a, w, h;
    double bmi;

    Console.Write("Enter age: ");
    a = Int32.Parse(Console.ReadLine());
    if (a < 18) {
        Console.WriteLine("Invalid age");
    }
    else {
        Console.Write("Enter weight in pounds: ");
        w = Int32.Parse(Console.ReadLine());
        Console.Write("Enter height in inches: ");
        h = Int32.Parse(Console.ReadLine());

        bmi = w * 703.0 / Math.Pow(h, 2);

        if (bmi < 15) {
            Console.WriteLine("Very severely underweight");
        }
        else if (bmi < 16) {
            Console.WriteLine("Severely underweight");
        }
        else if (bmi < 18.5) {
            Console.WriteLine("Underweight");
        }
        else if (bmi < 25) {
            Console.WriteLine("Normal");
        }
        else if (bmi < 30) {
            Console.WriteLine("Overweight");
        }
        else if (bmi < 35) {
            Console.WriteLine("Severely overweight");
        }
        else {
            Console.WriteLine("Very severely overweight");
        }
    }
    Console.ReadKey();
}
```

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*

```
double i;

i = 30.0;
while (i > 5) {
    Console.WriteLine(i);
    i /= 2;
}
Console.WriteLine("The end");
```

2. Solution

```
int i = 3;

do {
    i--;
} 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. 11.5

8. Solution

i. 4

ii. -2

iii. 2

iv. 10

9. Solution

```
static void Main() {
    double x, total;

    total = 0;

    i = 1;
    while (i <= 20) {
        Console.Write("Enter a number: ");
        x = Double.Parse(Console.ReadLine());
        if (x > 0) {
            total += x;
        }
        i += 1;
    }
    Console.WriteLine(total);
    Console.ReadKey();
}
```

10. Solution

```
static void Main() {
    int n, i;
    double p, x;

    Console.Write("Enter N: ");
    n = Int32.Parse(Console.ReadLine());

    p = 1;
    i = 1;
    while (i <= n ) {
        Console.Write("Enter a number: ");
        x = Double.Parse(Console.ReadLine());
        if (x > 0) {
            p *= x;
        }
        i += 1;
    }
    Console.WriteLine(p);
    Console.ReadKey();
}
```

11. Solution

```
static void Main() {
    int i, x, total;

    total = 0;

    i = 1;
    while (i <= 10 ) {
        Console.Write("Enter an integer: ");
        x = Int32.Parse(Console.ReadLine());
        if (x >= 100 && x <= 200) {
            total += x;
        }
        i += 1;
    }
    Console.WriteLine(total);
    Console.ReadKey();
}
```

12. Solution

```
static void Main() {
    int i, x, total;

    total = 0;
    i = 1;
    while (i <= 20 ) {
        Console.Write("Enter an integer between: ");
        x = Int32.Parse(Console.ReadLine());
        if (x >= 100 && x <= 999) {
            total += x;
        }
        i += 1;
    }
    Console.WriteLine(total);
    Console.ReadKey();
}
```

13. Solution

```
static void Main() {
    double x, p;

    p = 1;
    Console.Write("Enter a number: ");
    x = Double.Parse(Console.ReadLine());
    while (x != 0 ) {
        p *= x;
        Console.Write("Enter a number: ");
        x = Double.Parse(Console.ReadLine());
    }
    Console.WriteLine(p);
    Console.ReadKey();
}
```

14. Solution

```
static void Main() {
    double population;
    int years;

    population = 50000;

    years = 0;
    while (population >= 20000) {
        population -= population * 0.10;
        years++;
    }
    Console.Write(years);
    Console.ReadKey();
}
```

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. False
10. False

18.4 Review Questions: Multiple Choice

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

iii. -7 or -8

iv. -1

4. Solution

```
static void Main() {
    int i;
    double x, p, total;

    p = 1;
    total = 0;
    for (i = 1; i <= 20; i++) {
        Console.Write("Enter a number: ");
        x = Double.Parse(Console.ReadLine());
        p *= x;
        total += x;

    }
    Console.WriteLine(p + " " + total / 20);
    Console.ReadKey();
}
```

5. Solution

```
static void Main() {
    int n, i, count, x;

    Console.Write("Enter N: ");
    n = Int32.Parse(Console.ReadLine());

    count = 0;
    for (i = 1; i <= n; i++) {
        Console.Write("Enter an integer: ");
        x = Int32.Parse(Console.ReadLine());
        if (x > 0) {
            count += 1;
        }
    }

    if (count > 0) {
        Console.WriteLine(count);
    }
    else {
        Console.WriteLine("You entered no positive integers");
    }
    Console.ReadKey();
}
```

6. Solution

```
static void Main() {
    int i, start, finish;

    Console.Write("Enter value for start: ");
    start = Int32.Parse(Console.ReadLine());
    Console.Write("Enter value for finish: ");
    finish = Int32.Parse(Console.ReadLine());

    for (i = start; i <= finish; i++) {
        Console.WriteLine(i);
    }
    Console.ReadKey();
}
```

7. Solution

```
static void Main() {
    int e, i;
    double p, b;

    Console.Write("Enter a value for base: ");
    b = Double.Parse(Console.ReadLine());
    Console.Write("Enter an integer for exponent: ");
    e = Int32.Parse(Console.ReadLine());

    p = 1;
    for (i = 1; i <= e; i++) {
        p *= b;
    }

    Console.WriteLine(p);
    Console.ReadKey();
}
```

8. Solution

First Approach

```
static void Main() {
    string msg;
    int characters, i, count, words;

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

    characters = msg.Length;
    count = 0;
    for (i = 0; i <= characters - 1; i++) {
        if (msg[i].ToString() == " ") {
            count += 1;
        }
    }
    words = count + 1;

    Console.WriteLine("The message entered contains " + words + " words");
    Console.ReadKey();
}
```

Second Approach

```
static void Main() {
    string msg;
    int count, words;

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

    count = 0;
    foreach (var ch in msg) {
        if (ch.ToString() == " ") {
            count += 1;
        }
    }
    words = count + 1;

    Console.WriteLine("The message entered contains " + words + " words");
    Console.ReadKey();
}
```

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 between 45 and 49

iii. -7 or -8

iv. 138 or 139

2. Solution

```
static void Main() {
    int hour, minutes;

    for (hour = 0; hour <= 23; hour++) {
        for (minutes = 0; minutes <= 59; minutes++) {
            Console.WriteLine(hour + "\t " + minutes);
        }
    }
    Console.ReadKey();
}
```

3. Solution

```
static void Main() {
    int i, j;

    for (i = 5; i >= 1; i -= 1) {
        for (j = 1; j <= i; j++) {
            Console.Write(i);
        }
        Console.WriteLine();
    }
    Console.ReadKey();
}
```

4. Solution

```
static void Main() {
    int i, j;

    for (i = 1; i <= 6; i++) {
        for (j = 0; j <= i - 1; j++) {
            Console.Write(j);
        }
        Console.WriteLine();
    }
    Console.ReadKey();
}
```

5. Solution

```
static void Main() {
    int i, j;

    for (i = 1; i <= 3; i++) {
        for (j = 1; j <= 8; j++) {
            Console.Write("* ");
        }
        Console.WriteLine();
    }
    Console.ReadKey();
}
```

6. Solution

```
static void Main() {
    int n, i, j;

    Console.Write("Enter an integer between 3 and 20: ");
    n = Int32.Parse(Console.ReadLine());

    for (i = 1; i <= n; i++) {
        for (j = 1; j <= n; j++) {
            Console.Write("* ");
        }
        Console.WriteLine();
    }
    Console.ReadKey();
}
```

7. *Solution*

```
static void Main() {
    int i, j;

    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= i; j++) {
            Console.Write("* ");
        }
        Console.WriteLine();
    }
    Console.ReadKey();
}
```

8. *Solution*

```
static void Main() {
    int i, j;

    for (i = 1; i <= 5; i++) {
        for (j = 1; j <= i; j++) {
            Console.Write("* ");
        }
        Console.WriteLine();
    }

    for (i = 4; i >= 1; i -= 1) {
        for (j = 1; j <= i; j++) {
            Console.Write("* ");
        }
        Console.WriteLine();
    }
    Console.ReadKey();
}
```

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_not_johns = count_names = 0;
~~name = "";~~
Console.Write("Enter a name: ");
name = Console.ReadLine();
while (name != "STOP") {
    ~~Console.Write("Enter a name: ");~~
    ~~name = Console.ReadLine();~~
    count_names += 1;
    if (name != "John") {
        count_not_johns += 1;
    }
    Console.Write("Enter a name: ");
    name = Console.ReadLine();
}

Console.WriteLine("Names other than John: " + count_ not_johns);
Console.WriteLine(count_names + " names entered");
Console.ReadKey();
```

2. *Solution*

```
static void Main() {
    string text;
    bool found;

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

    found = false;
    foreach (var ch in text) {
        if (ch.ToString() == " ") {
            found = true;
            break;
        }
    }

    if (found == false) {
        Console.WriteLine("One Single Word");
    }
    else {
        Console.WriteLine("Complete Sentence");
    }
    Console.ReadKey();
}
```

3. Solution

```
static void Main() {
    string sentence;

    bool found;
    Console.Write("Enter a sentence: ");
    sentence = Console.ReadLine();

    found = false;
    foreach (var ch in "0123456789") {
        if (sentence.IndexOf(ch) > -1) {
            found = true;
            break;
        }
    }

    if (found == true) {
        Console.WriteLine("The sentence contains a number");
    }
    Console.ReadKey();
}
```

4. Solution

```
Console.WriteLine("Printing all integers from 1 to 100");
i = 1;
while (i < 101) {
    Console.WriteLine(i++);
}
```

5. Solution

```
static void Main() {
Console.WriteLine("Printing odd integers from 1 to 99");
i = 1;
while (!(i > 100)) {
    Console.WriteLine(i);
    i += 2;
}
```

6. *Solution*

```
static void Main() {
    int i, j;

    for (i = 1; i <= 4; i++) {
        for (j = 1; j <= 4; j++) {
            Console.WriteLine(i + " x " + j + " = " + (i * j));
        }
    }
    Console.ReadKey();
}
```

7. *Solution*

```
static void Main() {
    int i, j;

    Console.Write("\t|\t");
    for (i = 1; i <= 12; i++) {
        Console.Write(i + "\t");
    }
    Console.WriteLine();

    for (i = 1; i <= 12; i++) {
        Console.Write("---------");
    }
    Console.WriteLine();

    for (i = 1; i <= 12; i++) {
        Console.Write(i + "\t|\t");
        for (j = 1; j <= 12; j++) {
            Console.Write(i * j + "\t");
        }
        Console.WriteLine();
    }
    Console.ReadKey();
}
```

Chapter 21

21.2 Review Exercises

1. *Solution*

```
static void Main() {
    int i, total;

    total = 0;
    for (i = 1; i <= 99; i += 2) {
        total += i;
    }

    Console.WriteLine(total);
    Console.ReadKey();
}
```

2. *Solution*

```
static void Main() {
    int n, total, i;

    Console.Write("Enter N: ");
    n = Int32.Parse(Console.ReadLine());

    total = 0;
    for (i = 2; i <= 2 * n; i += 2) {
        total += i;
    }

    Console.WriteLine(total);
    Console.ReadKey();
}
```

3. *Solution*

```
static void Main() {
    int n, i, grade, total, count;

    Console.Write("Enter total number of students: ");
    n = Int32.Parse(Console.ReadLine());

    total = 0;
    count = 0;
    for (i = 1; i <= n; i++) {
        Console.Write("Enter grade: ");
        grade = Int32.Parse(Console.ReadLine());
        if (grade >= 90 && grade <= 100) {
            total += grade;
            count += 1;
        }
    }

    if (count > 0) {
        Console.WriteLine(total / (double)count);
    }
    else {
        Console.WriteLine("There are no students that got an A");
    }
    Console.ReadKey();
}
```

4. Solution

```
static void Main() {
    int count_pos, count_neg, total_pos, total_neg, i, x;

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

    for (i = 1; i <= 50; i++) {
        Console.Write("Enter an integer: ");
        x = Int32.Parse(Console.ReadLine());
        if (x > 0) {
            count_pos += 1;
            total_pos += x;
        }
        else if (x < 0) {
            count_neg += 1;
            total_neg += x;
        }
    }

    if (count_pos > 0) {
        Console.WriteLine(total_pos / (double)count_pos);
    }

    if (count_neg > 0) {
        Console.WriteLine(total_neg / (double)count_neg);
    }
    Console.ReadKey();
}
```

5. Solution

```
static void Main() {
    int count;
    double total, x;

    total = 0;
    count = 0;
    do {
        Console.Write("Enter a number: ");
        x = Double.Parse(Console.ReadLine());
        if (x == 0) {
            count += 1;
        }
        total += x;
    } while (total <= 3000);

    Console.WriteLine(count);
    Console.ReadKey();
}
```

6. Solution

```
static void Main() {
    string answer;
    double r, area;

    do {
        Console.Write("Enter the length of a radius: ");
        r = Double.Parse(Console.ReadLine());

        area = 3.141 * Math.Pow(r, 2);
        Console.WriteLine("The area is: " + area);

        Console.Write("Would you like to repeat? ");
        answer = Console.ReadLine();
    } while (answer.ToUpper() == "YES");
}
```

7. Solution

```
static void Main() {
    long x;

    x = 1;
    while (x <= 1073741824) {
        Console.WriteLine(x);
        x *= 2;
   }
    Console.ReadKey();
}
```

8. Solution

```
static void Main() {
    int i;

    for (i = 1; i <= 100; i++) {
        Console.WriteLine(-i + "\n " + i);
    }
    Console.ReadKey();
}
```

9. Solution

First approach

```
static void Main() {
    int i;
    double offset, value;

    value = 0;
    for (i = 0; i <= 7; i++) {
        offset = Math.Pow(10, i);
        value += offset;
        Console.WriteLine(value);
    }
    Console.ReadKey();
}
```

Second approach

```
static void Main() {
    string value;
    int i;

    value = "1";
    for (i = 0; i <= 7; i++) {
        Console.WriteLine(value);
        value += "1";
    }
    Console.ReadKey();
}
```

10. Solution

```
static void Main() {
    double t, maximum, total;
    int i;

    Console.Write("Enter temperature for day 1: ");
    t = Double.Parse(Console.ReadLine());
    maximum = t;
    total = t;
    for (i = 2; i <= 31; i++) {
        Console.Write("Enter temperature for day " + i + ": ");
        t = Double.Parse(Console.ReadLine());

        total += t;
        if (t > maximum) {
            maximum = t;
        }
    }
    Console.WriteLine(total / 31 + " " + maximum);
    Console.ReadKey();
}
```

11. Solution

```
static void Main() {
    double level, maximum, minimum;
    int hour, min_hour, max_hour, i;

    Console.Write("Enter level: ");
    level = Double.Parse(Console.ReadLine());
    Console.Write("Enter hour: ");
    hour = Int32.Parse(Console.ReadLine());

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

    for (i = 2; i <= 24; i++) {
        Console.Write("Enter level: ");
        level = Double.Parse(Console.ReadLine());
        Console.Write("Enter hour: ");
        hour = Int32.Parse(Console.ReadLine());

        if (level > maximum) {
            maximum = level;
            max_hour = hour;
        }

        if (level < minimum) {
            minimum = level;
            min_hour = hour;
        }
    }
    Console.WriteLine(maximum + " " + max_hour);
    Console.WriteLine(minimum + " " + min_hour);
    Console.ReadKey();
}
```

12. Solution

```
static void Main() {
    int attempts = 0, first_player_attempts = 0, second_player_attempts = 0;
    int guess, i, secret_number;

    Random rnd = new Random(Guid.NewGuid().GetHashCode());

    for (i = 1; i <= 2; i++) {
        secret_number = rnd.Next(1, 101);

        attempts = 1;
        Console.Write("Player " + i + ", enter a guess: ");
        guess = Int32.Parse(Console.ReadLine());
        while (guess != secret_number) {
            if (guess > secret_number) {
                Console.Write("Your guess is bigger than my secret number.");
            }
            else {
                Console.Write("Your guess is smaller than my secret number.");
            }
            attempts++;
            Console.Write(" Try again! Enter a guess: ");
            guess = Int32.Parse(Console.ReadLine());
        }
        Console.WriteLine("You found it!");
        Console.WriteLine("Attempts: " + attempts);

        if (i == 0) {
            first_player_attempts = attempts;
        }
        else {
            second_player_attempts = attempts;
        }
    }

    if (first_player_attempts < second_player_attempts) {
        Console.WriteLine("First player wins!");
    }
    else if (first_player_attempts > second_player_attempts) {
        Console.WriteLine("Second player wins!");
    }
    else {
        Console.WriteLine("It's a draw");
    }
    Console.ReadKey();
}
```

13. Solution

```
static void Main() {
    string gender;
    int n, i, grade, total, total_a, count_a, total_b, count_b;
    int total_a_boys, count_a_boys, count_cdef_girls;

    Console.Write("Enter total number of students: ");
    n = Int32.Parse(Console.ReadLine());

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

    for (i = 1; i <= n; i++) {
        Console.Write("Enter grade for student No " + i + ": ");
        grade = Int32.Parse(Console.ReadLine());

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

        if (grade >= 90 && grade <= 100) {
            total_a += grade;
            count_a += 1;
            if (gender == "M") {
                total_a_boys += grade;
                count_a_boys += 1;
            }
        }
        else if (grade >= 80 && grade <= 89) {
            total_b += grade;
            count_b += 1;
        }
        else {
            if (gender == "F") {
                count_cdef_girls += 1;
            }
        }
        total += grade;
    }

    if (count_a > 0) {
        Console.Write("Average value of those who got an 'A': ");
        Console.WriteLine(total_a / (double)count_a);
    }

    if (count_b > 0) {
        Console.Write("Average value of those who got a 'B': ");
        Console.WriteLine(total_b / (double)count_b);
    }

    if (count_a_boys > 0) {
        Console.Write("Average value of boys who got an 'A': ");
        Console.WriteLine(total_a_boys / (double)count_a_boys);
    }

    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 / (double)n);
    Console.ReadKey();
}
```

14. Solution

```
static void Main() {
    string answer;
    double amount, discount, amount_to_pay;

    do {
        Console.Write("Enter amount: ");
        amount = Double.Parse(Console.ReadLine());

        if (amount < 20) {
            discount = 0;
        }
        else if (amount < 50) {
            discount = 3;
        }
        else if (amount < 100) {
            discount = 5;
        }
        else {
            discount = 10;
        }

        Console.WriteLine("Discount: " + discount + "%");

        Console.Write("Would you like to repeat? ");
        answer = Console.ReadLine();
    } while (answer.ToUpper() == "YES");
}
```

Chapter 22

22.12 Review Exercises

1. *Solution*

```
private void Form1_Shown(object sender, EventArgs e) {
    Turtle.Forward(150);
    Turtle.Angle = -130;
    Turtle.Forward(50);

    Turtle.X = 0;
    Turtle.Y = 150;

    Turtle.Angle = 130;
    Turtle.Forward(50);
}
```

2. *Solution*

```
private void Form1_Shown(object sender, EventArgs e) {
    Turtle.Forward(100);
    Turtle.Rotate(90);
    Turtle.Forward(200);
    Turtle.Rotate(90);
    Turtle.Forward(100);
    Turtle.Rotate(90);
    Turtle.Forward(200) ;
}
```

3. *Solution*

First approach

```
private void Form1_Shown(object sender, EventArgs e) {
    Turtle.Rotate(30);
    Turtle.Forward(100);
    Turtle.Rotate(120);
    Turtle.Forward(100);
    Turtle.Rotate(60);
    Turtle.Forward(100);
    Turtle.Rotate(120);
    Turtle.Forward(100) ;
}
```

Second approach

```
private void Form1_Shown(object sender, EventArgs e) {
    Turtle.Angle = 30;
    Turtle.Forward(100);
    Turtle.Angle = 150;
    Turtle.Forward(100);
    Turtle.Angle = 210;
    Turtle.Forward(100);
    Turtle.Angle = 330;
    Turtle.Forward(100) ;
}
```

4. Solution

```
private void Form1_Shown(object sender, EventArgs e) {
    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) ;
}
```

5. Solution

```
private void Form1_Shown(object sender, EventArgs e) {
    int k, i;

    Turtle.PenSize = 2;
    for (k = 1; k <= 2; k++) {
        for (i = 1; i <= 4; i++) {
            Turtle.Forward(100);
            Turtle.Rotate(90);
        }
        Turtle.X -= 100;
    }
}
```

6. Solution

```
private void Form1_Shown(object sender, EventArgs e) {
    int m, n, i;

    for (m = 1; m <= 2; m++) {
        for (n = 1; n <= 2; n++) {
            for (i = 1; i <= 4; i++) {
                Turtle.Forward(100);
                Turtle.Rotate(90);
            }
            Turtle.X -= 150;
        }
        Turtle.X = 0;
        Turtle.Y -= 150;
    }
}
```

7. Solution

```
private void Form1_Shown(object sender, EventArgs e) {
    int size, length, height;

    size = 2;
    length = 250;
    height = 150;

    Turtle.PenSize = size;
    Turtle.Forward(height);
    Turtle.Rotate(90);
    Turtle.Forward(length);
    Turtle.Rotate(90);
    Turtle.Forward(height);
    Turtle.Rotate(90);
    Turtle.Forward(length);
}
```

8. Solution

```
private void Form1_Shown(object sender, EventArgs e) {
    int length = 150;

    Turtle.Angle = 90;

    Turtle.Forward(length);
    Turtle.Rotate(-120);
    Turtle.Forward(length);
    Turtle.Rotate(-120);
    Turtle.Forward(length);
    Turtle.Rotate(-120);
    Turtle.Forward(length) ;
}
```

9. Solution

```
private void Form1_Shown(object sender, EventArgs e) {
    int i;

    for (i = 1; i <= 12; i++) {
        Turtle.Forward(100);
        Turtle.X = 0;
        Turtle.Y = 0;
        Turtle.Rotate(30);
    }
}
```

10. Solution

```
private void Form1_Shown(object sender, EventArgs e) {
    int k, i;

    Turtle.PenSize = 2;
    Turtle.Angle = 90;
    for (k = 1; k <= 180; k += 60) {
        for (i = 1; i <= 5; i++) {
            Turtle.Forward(150 + k);
            Turtle.Rotate(180f / 5 * 4);
        }
        Turtle.X -= 30;
        Turtle.Y += 10;
    }
}
```

11. Solution

```
private void Form1_Shown(object sender, EventArgs e) {
    int k, i;
    Turtle.PenSize = 2;
    for (k = 1; k <= 3; k++) {
        for (i = 1; i <= 4; i++) {
            Turtle.Forward(100);
            Turtle.Rotate(90);
        }
        Turtle.Rotate(-30);
    }
}
```

12. Solution

```
private void Form1_Shown(object sender, EventArgs e) {
    int k, i;
    Turtle.PenSize = 2;
    for (k = 1; k <= 12; k++) {
        for (i = 1; i <= 4; i++) {
            Turtle.Forward(100);
            Turtle.Rotate(90);
        }
        Turtle.Rotate(-30);
    }
}
```

13. Solution

```
private void Form1_Shown(object sender, EventArgs e) {
    int k, i;
    Turtle.PenSize = 2;
    for (k = 1; k <= 8; k++) {
        for (i = 1; i <= 4; i++) {
            Turtle.Forward(100);
            Turtle.Rotate(90);
        }
        Turtle.Rotate(-45);
    }
}
```

14. Solution

```
private void Form1_Shown(object sender, EventArgs e) {
    int i, k;

    Turtle.PenSize = 1;

    //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.Angle = 45;
    Turtle.PenColor = Color.Red;
    Turtle.Forward(141);
    Turtle.Rotate(90);
    Turtle.Forward(141);

    //Draw the windows
    Turtle.PenColor = Color.Brown;
    Turtle.X = 20;
    Turtle.Y = 50;

    for (k = 0; k <= 1; k++) {
        Turtle.Angle = 0;

        for (i = 0; i <= 3; i++) {
            Turtle.Forward(40);
            Turtle.Rotate(90);
        }

        Turtle.Angle = 0;
        Turtle.X += 20;
        Turtle.Forward(40);

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

        Turtle.X += 80;
        Turtle.Y -= 20;
    }

    //Draw the door
    Turtle.X -= 180;
    Turtle.Y = 0;
    Turtle.Angle = 0;

    Turtle.Forward(70);
    Turtle.Rotate(90);
    Turtle.Forward(40);
    Turtle.Rotate(90);
    Turtle.Forward(70);
    Turtle.Rotate(90);
    Turtle.Forward(40);
}
```

15. Solution

```
//Move to poll position
Turtle.X = -300;

for (m = 1; m <= 3; m++) {
    //Here goes the code of the previous exersise

    Turtle.Angle = 0;
    Turtle.X += 200;
}
```

The final program becomes

```
private void Form1_Shown(object sender, EventArgs e) {
    int i, m, k;

    Turtle.PenSize = 1;

    //Move to poll position
    Turtle.X = -300;
```

```
for (m = 1; m <= 3; m++) {
    //Draw a blue rectangle
    Turtle.PenColor = Color.Blue;
    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.Angle = 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; k <= 1; k++) {
        Turtle.Angle = 0;

        for (i = 0; i <= 3; i++) {
            Turtle.Forward(40);
            Turtle.Rotate(90);
        }

        Turtle.Angle = 0;
        Turtle.X += 20;
        Turtle.Forward(40);

        Turtle.Angle = 90;
        Turtle.X -= 20;
        Turtle.Y = 70;
        Turtle.Forward(40);
```

```
            Turtle.X += 80;
            Turtle.Y -= 20;
        }

        //Draw the door
        Turtle.X -= 180;
        Turtle.Y = 0;
        Turtle.Angle = 0;

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

        Turtle.Angle = 0;
        Turtle.X += 200;
    }
}
```

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. False
36. True
37. True
38. True
39. True
40. True
41. True
42. False
43. False
44. True
45. False

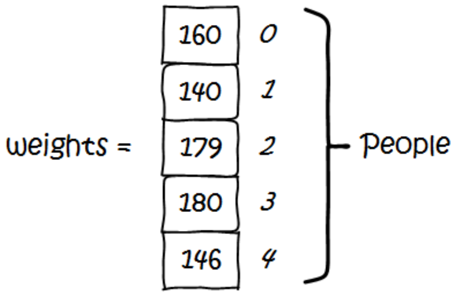
23.15 Review Questions: Multiple Choice

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

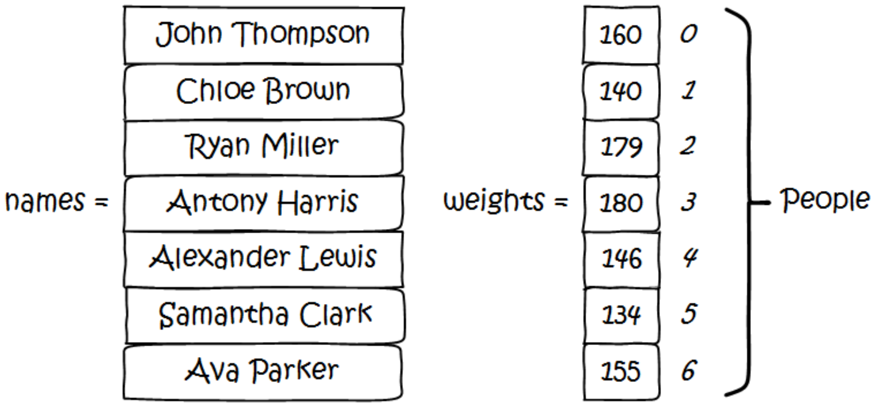
9. b
10. b
11. c
12. a
13. a
14. b
15. b
16. a
17. b

23.16 Review Exercises

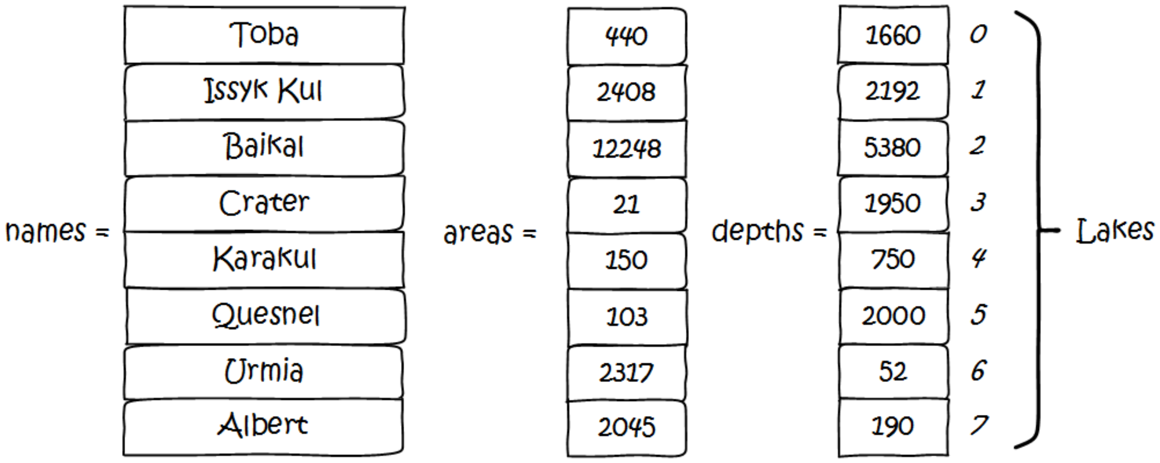
1. Solution



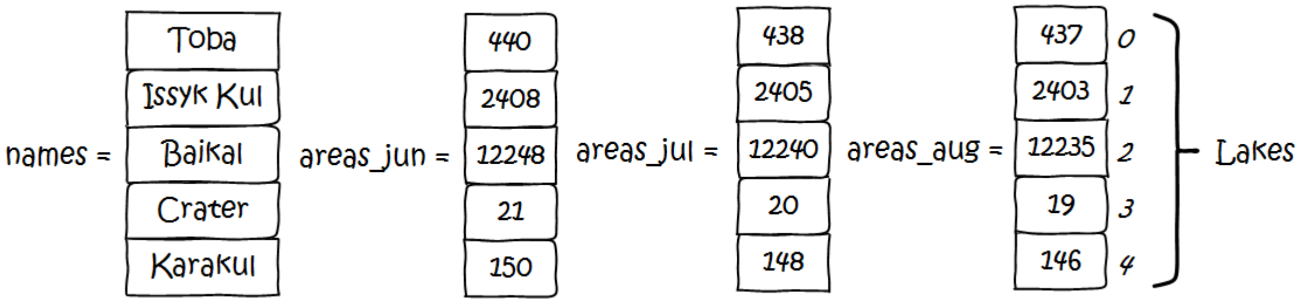
2. Solution



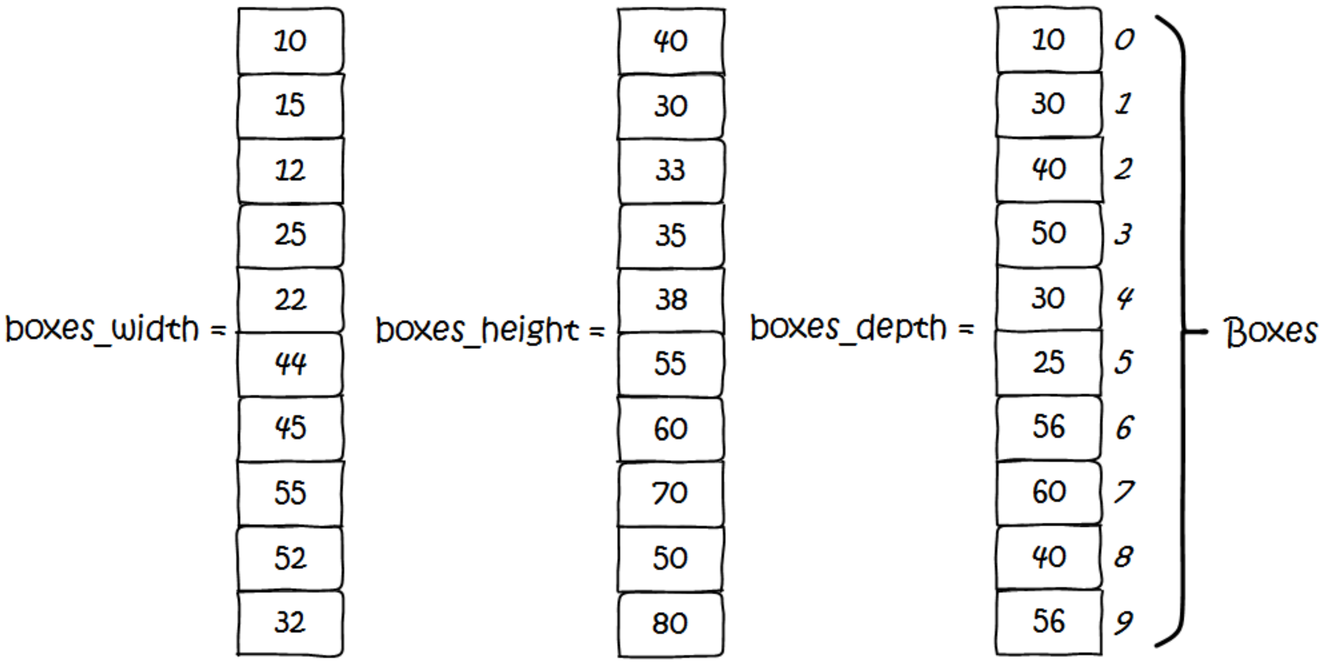
3. Solution



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 int ELEMENTS = 100;
static void Main() {
    int i;
    double[] a = new double[ELEMENTS];

    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.WriteLine(Math.Pow(a[i], 3));
    }

    Console.ReadKey();
}
```

12. Solution

```
const int ELEMENTS = 80;
static void Main() {
    int i;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    for (i = 0; i <= ELEMENTS - 1; i++) {
        a[i] = Math.Pow(a[i], 2);
    }

    for (i = ELEMENTS - 1; i >= 0; i -= 1) {
        Console.WriteLine(a[i]);
    }
    Console.ReadKey();
}
```

13. Solution

```
const int ELEMENTS = 50;
static void Main() {
    int i;

    int[] a = new int[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter an integer: ");
        a[i] = Int32.Parse(Console.ReadLine());
    }

    foreach (var element in a) {
        if (element >= 10) {
            Console.WriteLine(element);
        }
    }
    Console.ReadKey();
}
```

14. Solution

```
const int ELEMENTS = 30;
static void Main() {
    int i;
    double total;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    total = 0;
    foreach (var element in a) {
        if (element > 0) {
            total += element;
        }
    }
    Console.WriteLine(total);
    Console.ReadKey();
}
```

15. Solution

```
const int ELEMENTS = 50;
static void Main() {
    int i, total;

    int[] a = new int[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter an integer: ");
        a[i] = Int32.Parse(Console.ReadLine());
    }

    total = 0;
    foreach (var element in a) {
        if (element >= 10 && element <= 99) {
            total += element;
        }
    }
    Console.WriteLine(total);
    Console.ReadKey();
}
```

16. Solution

```
const int ELEMENTS = 40;
static void Main() {
    int i;
    double total_pos, total_neg;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    total_pos = 0;
    total_neg = 0;
    foreach (var element in a) {
        if (element > 0) {
            total_pos += element;
        }
        else if (element < 0) {
            total_neg += element;
        }
    }

    Console.WriteLine(total_pos + " " + total_neg);
    Console.ReadKey();
}
```

17. Solution

First approach

```
const int ELEMENTS = 20;
static void Main() {
    int i;
    double total;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    total = 0;
    foreach (var element in a) {
        total += element;
    }

    Console.WriteLine(total / ELEMENTS);
    Console.ReadKey();
}
```

Second approach

```
const int ELEMENTS = 20;
static void Main() {
    int i;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    Console.WriteLine(a.Sum() / ELEMENTS);
    Console.ReadKey();
}
```

18. Solution

```
const int ELEMENTS = 50;
static void Main() {
    int i;

    int[] a = new int[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter an integer: ");
        a[i] = Int32.Parse(Console.ReadLine());
    }

    for (i = 0; i <= ELEMENTS - 1; i++) {
        if (a[i] < 20) {
            Console.WriteLine(i);
        }
    }
    Console.ReadKey();
}
```

19. Solution

```
const int ELEMENTS = 60;
static void Main() {
    int i;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    for (i = 0; i <= ELEMENTS - 1; i += 2) {
        Console.WriteLine(a[i]);
    }
    Console.ReadKey();
}
```

20. Solution

```
const int ELEMENTS = 20;
static void Main() {
    int i;
    double total;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    total = 0;
    for (i = 0; i <= ELEMENTS - 1; i += 2) {
        total += a[i];
    }

    Console.WriteLine(total);
    Console.ReadKey();
}
```

21. Solution

First approach

```
int[] a = new int[100];
for (i = 0; i <= 99; i++) {
    a[i] = i + 1;
}
```

Second approach

```
int[] a = new int[100];
for (i = 1; i <= 100; i++) {
    a[i - 1] = i;
}
```

22. Solution

First approach

```
int[] a = new int[100];
for (i = 0; i <= 9; i++) {
    a[i] = 2 * (i + 1);
}
```

Second approach

```
int[] a = new int[100];
for (i = 1; i <= 100; i++) {
    a[i - 1] = 2 * i;
}
```

23. Solution

```
static void Main() {
    int i, n;

    Console.Write("Enter N: ");
    n = Int32.Parse(Console.ReadLine());

    double[] a = new double[n];
    for (i = 1; i <= n; i++) {
        a[i - 1] = Math.Pow(i, 2);
    }

    foreach (var element in a) {
        Console.WriteLine(element);
    }
    Console.ReadKey();
}
```

24. Solution

```
const int ELEMENTS = 10;
static void Main() {
    int i;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    for (i = 0; i <= ELEMENTS - 1; i++) {
        if (a[i] == (int)a[i]) {
            Console.WriteLine(i);
        }
    }
    Console.ReadKey();
}
```

25. Solution

```
const int ELEMENTS = 50;
```

```
static void Main() {
    int i, count;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    count = 0;
    for (i = 0; i <= ELEMENTS - 1; i++) {
        if (a[i] < 0) {
            count += 1;
        }
    }

    Console.WriteLine(count);
    Console.ReadKey();
}
```

26. Solution

```
const int ELEMENTS = 20;
static void Main() {
    int i;

    string[] words = new string[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a word: ");
        words[i] = Console.ReadLine();
    }

    foreach (var word in words) {
        if (word.Length < 5) {
            Console.WriteLine(word);
        }
    }
    Console.ReadKey();
}
```

27. Solution

```
const int ELEMENTS = 30;
static void Main() {
    int i;

    string[] words = new string[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a word: ");
        words[i] = Console.ReadLine();
    }

    int[] length_limits = { 5, 10, 20 };

    foreach (var length_limit in length_limits) {
        foreach (var word in words) {
            if (word.Length < length_limit) {
                Console.WriteLine(word);
            }
        }
    }
    Console.ReadKey();
}
```

28. Solution

```
const int ELEMENTS = 40;
static void Main() {
    int i, count;

    string[] words = new string[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("Enter a word: ");
        words[i] = Console.ReadLine();
    }

    foreach (var word in words) {
        count = 0;
        foreach (var letter in word) {
            if (letter.ToString() == "w") {
                count += 1;
            }

            if (count == 2) {
                Console.WriteLine(word);
                break;
            }
        }
    }
    Console.ReadKey();
}
```

Chapter 24

24.5 Review Questions: True/False

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

24.6 Review Exercises

1. Solution

```
const int ELEMENTS_OF_A = 50;
const int ELEMENTS_OF_NEW = ELEMENTS_OF_A - 2;
static void Main() {
    int i;

    double[] a = new double[ELEMENTS_OF_A];
    for (i = 0; i <= ELEMENTS_OF_A - 1; i++) {
        Console.Write("Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    double[] new_arr = new double[ELEMENTS_OF_NEW];
    for (i = 0; i <= ELEMENTS_OF_NEW - 1; i++) {
        new_arr[i] = (a[i] + a[i + 1] + a[i + 2]) / 3;
    }

    for (i = 0; i <= ELEMENTS_OF_NEW - 1; i++) {
        Console.WriteLine(new_arr[i]);
    }
    Console.ReadKey();
}
```

2. Solution

```
const int ELEMENTS = 15;
static void Main() {
    int i;
    double minimum;

    double[] a = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("A - Enter a number: ");
        a[i] = Double.Parse(Console.ReadLine());
    }

    double[] b = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("B - Enter a number: ");
        b[i] = Double.Parse(Console.ReadLine());
    }

    double[] c = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.Write("C - Enter a number: ");
        c[i] = Double.Parse(Console.ReadLine());
    }

    double[] new_arr = new double[ELEMENTS];
    for (i = 0; i <= ELEMENTS - 1; i++) {
        minimum = a[i];
        if (b[i] < minimum) {
            minimum = b[i];
        }
        if (c[i] < minimum) {
            minimum = c[i];
        }
        new_arr[i] = minimum;
    }

    for (i = 0; i <= ELEMENTS - 1; i++) {
        Console.WriteLine(new_arr[i]);
    }
    Console.ReadKey();
}
```

3. Solution

```
const int MOUNTAINS = 30;
static void Main() {
    int i, index_of_max, index_of_min;
    double maximum, minimum;

    string[] names = new string[MOUNTAINS];
    double[] heights = new double[MOUNTAINS];
    string[] countries = new string[MOUNTAINS];
    for (i = 0; i <= MOUNTAINS - 1; i++) {
        names[i] = Console.ReadLine();
        heights[i] = Double.Parse(Console.ReadLine());
        countries[i] = Console.ReadLine();
    }

    maximum = heights[0];
    index_of_max = 0;
    minimum = heights[0];
    index_of_min = 0;
    for (i = 1; i <= MOUNTAINS - 1; i++) {
        if (heights[i] > maximum) {
            maximum = heights[i];
            index_of_max = i;
        }
        if (heights[i] < minimum) {
            minimum = heights[i];
            index_of_min = i;
        }
    }

    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]);
    Console.ReadKey();
}
```

4. Solution

```
const int CLASS1 = 20;
const int CLASS2 = 25;
static void Main() {
    int i;
    string needle;
    bool found;

    Console.WriteLine("Class A");
    string[] names1 = new string[CLASS1];
    for (i = 0; i <= CLASS1 - 1; i++) {
        Console.Write("Enter name: ");
        names1[i] = Console.ReadLine();
    }

    Console.WriteLine("Class B");
    string[] names2 = new string[CLASS2];
    for (i = 0; i <= CLASS2 - 1; i++) {
        Console.Write("Enter name: ");
        names2[i] = Console.ReadLine();
    }

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

    found = false;
    foreach (var name in names1) {
        if (name == needle) {
            found = true;
            break;
        }
    }
    if (found == true) {
        Console.WriteLine("Student found in class No 1");
    }
    else {
        found = false;
        foreach (var name in names2) {
            if (name == needle) {
                found = true;
                break;
            }
        }

        if (found == true) {
            Console.WriteLine("Student found in class No 2");
        }
        else {
            Console.WriteLine("Student not found in either class");
        }
    }
    Console.ReadKey();
}
```

5. Solution

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

found = false;
for (i = 0; i <= 99; i++) {
    if (usernames[i] == usr) {
        found = true;
        break;
    }
}

if (found == true) {
    if (usernames[i] == usr && passwords[i] == pwd) {
        Console.WriteLine("Login OK!");
    }
    else {
        Console.WriteLine("Login Failed!");
    }
}
else {
    Console.WriteLine("Login Failed!");
}
```

6. Solution

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

found = false;
for (i = 0; i <= 999; i++) {
    if (SSNs[i] == needle) {
        found = true;
        Console.WriteLine(SSNs[i] + " " + names[i]);
        break;
    }
}

if (found == false) {
    for (i = 0; i <= 999; i++) {
        if (names[i] == needle) {
            found = true;
            Console.WriteLine(SSNs[i] + " " + names[i]);
        }
    }
}

if (found == false) {
    Console.WriteLine("This value does not exist");
}
```

7. Solution

```
const int STUDENTS = 12;
static void Main() {
    int i;
    bool found;

    int[] grades1 = new int[STUDENTS];
    int[] grades2 = new int[STUDENTS];
    int[] grades3 = new int[STUDENTS];
    for (i = 0; i <= STUDENTS - 1; i++) {
        grades1[i] = Int32.Parse(Console.ReadLine());
        grades2[i] = Int32.Parse(Console.ReadLine());
        grades3[i] = Int32.Parse(Console.ReadLine());
    }

    found = false;
    for (i = 0; i <= STUDENTS - 1; i++) {
        if ((grades1[i] + grades2[i] + grades3[i]) / 3.0 < 70) {
            found = true;
            break;
        }
    }

    if (found == true) {
        Console.Write("At least one student has an average value below 70");
    }
    Console.ReadKey();
}
```

8. Solution

```
const int STUDENTS = 15;
static void Main() {
    int i;
    double average;

    int[] grades1 = new int[STUDENTS];
    int[] grades2 = new int[STUDENTS];
    for (i = 0; i <= STUDENTS - 1; i++) {
         grades1[i] = Int32.Parse(Console.ReadLine());
         grades2[i] = Int32.Parse(Console.ReadLine());
    }

    for (i = 0; i <= STUDENTS - 1; i++) {
        Console.WriteLine("Student No " + (i + 1) + ": ");

        average = (grades1[i] + grades2[i]) / 2.0;

        if (average < 60) {
            Console.WriteLine("E/F");
        }
        else if (average < 70) {
            Console.WriteLine("D");
        }
        else if (average < 80) {
            Console.WriteLine("C");
        }
        else if (average < 90) {
            Console.WriteLine("B");
        }
        else {
            Console.WriteLine("A");
        }
    }
    Console.ReadKey();
}
```

9. Solution

```
const int PLAYERS = 15;
static void Main() {
    int i, total;

    int[] points_match1 = new int[PLAYERS];
    int[] points_match2 = new int[PLAYERS];
    int[] points_match3 = new int[PLAYERS];
    int[] points_match4 = new int[PLAYERS];
    for (i = 0; i <= PLAYERS - 1; i++) {
         points_match1[i] = Int32.Parse(Console.ReadLine());
         points_match2[i] = Int32.Parse(Console.ReadLine());
         points_match3[i] = Int32.Parse(Console.ReadLine());
         points_match4[i] = Int32.Parse(Console.ReadLine());
    }

    for (i = 0; i <= PLAYERS - 1; i++) {
        Console.WriteLine("Player No " + (i + 1));
        total = points_match1[i] + points_match2[i] +
                points_match3[i] + points_match4[i];
        Console.WriteLine(total);
    }
    Console.ReadKey();
}
```

10. Solution

```
const int HOURS = 24;
static void Main() {
    int i;
    double average;

    double[] t_city1 = new double[HOURS];
    double[] t_city2 = new double[HOURS];
    double[] t_city3 = new double[HOURS];
    for (i = 0; i <= HOURS - 1; i++) {
         t_city1[i] = Double.Parse(Console.ReadLine());
         t_city2[i] = Double.Parse(Console.ReadLine());
         t_city3[i] = Double.Parse(Console.ReadLine());
    }

    for (i = 0; i <= HOURS - 1; i++) {
        average = (t_city1[i] + t_city2[i] + t_city3[i]) / 3;
        if (average < 10) {
            Console.WriteLine("Hour: " + (i + 1));
        }
    }
    Console.ReadKey();
}
```

11. Solution

```
const int STUDENTS = 12;
static void Main() {
    int i;

    string[] names = new string[STUDENTS];
    int[] grd_lesson1 = new int[STUDENTS];
    int[] grd_lesson2 = new int[STUDENTS];
    for (i = 0; i <= STUDENTS - 1; i++) {
         names[i] = Console.ReadLine();
         grd_lesson1[i] = Int32.Parse(Console.ReadLine());
         grd_lesson2[i] = Int32.Parse(Console.ReadLine());
    }

    //create list average
    double[] average = new double[STUDENTS];
    for (i = 0; i <= STUDENTS - 1; i++) {
        average[i] = (grd_lesson1[i] + grd_lesson2[i]) / 2.0;
    }

    for (i = 0; i <= STUDENTS - 1; i++) {
        Console.WriteLine(names[i] + " " + average[i]);
    }

    for (i = 0; i <= STUDENTS - 1; i++) {
        if (average[i] < 60) {
            Console.WriteLine(names[i]);
        }
    }

    for (i = 0; i <= STUDENTS - 1; i++) {
        if (average[i] > 89) {
            Console.WriteLine(names[i] + " Bravo!");
        }
    }
    Console.ReadKey();
}
```

12. Solution

```
const int ARTISTS = 15;
static void Main() {
    int i, minimum;

    string[] artist_names = new string[ARTISTS];
    string[] song_titles = new string[ARTISTS];
    int[] scoreA = new int[ARTISTS];
    int[] scoreB = new int[ARTISTS];
    int[] scoreC = new int[ARTISTS];

    for (i = 0; i <= ARTISTS - 1; i++) {
        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] = Int32.Parse(Console.ReadLine());

        Console.Write(" gotten from judge B: ");
        scoreB[i] = Int32.Parse(Console.ReadLine());

        Console.Write(" gotten from judge C: ");
        scoreC[i] = Int32.Parse(Console.ReadLine());
    }

    int[] total = new int[ARTISTS];
    for (i = 0; i <= ARTISTS - 1; i++) {
        minimum = scoreA[i];
        if (scoreB[i] < minimum) {
            minimum = scoreB[i];
        }
        if (scoreC[i] < minimum) {
            minimum = scoreC[i];
        }

        total[i] = scoreA[i] + scoreB[i] + scoreC[i] - minimum;
    }

    for (i = 0; i <= ARTISTS - 1; i++) {
        Console.WriteLine(artist_names[i]);
        Console.WriteLine(song_titles[i] + ": " + total[i]);
    }
    Console.ReadKey();
}
```

13. Solution

```
const int CITIZENS = 20;
static void Main() {
    string prod_name1, prod_name2;
    int i, count_A;

    string[] answers1 = new string[CITIZENS];
    string[] answers2 = new string[CITIZENS];

    Console.Write("Enter Product Name 1: ");
    prod_name1 = Console.ReadLine();
    for (i = 0; i <= CITIZENS - 1; i++) {
        Console.Write("Enter score for product " + prod_name1 + ": ");
        answers1[i] = Console.ReadLine();
    }

    Console.Write("Enter Product Name 2: ");
    prod_name2 = Console.ReadLine();
    for (i = 0; i <= CITIZENS - 1; i++) {
        Console.Write("Enter score for product " + prod_name2 + ": ");
        answers2[i] = Console.ReadLine();
    }

    count_A = 0;
    for (i = 0; i <= CITIZENS - 1; i++) {
        if (answers1[i] == "A") {
            count_A += 1;
        }
    }
    Console.WriteLine(prod_name1 + " " + count_A);

    count_A = 0;
    for (i = 0; i <= CITIZENS - 1; i++) {
        if (answers2[i] == "A") {
            count_A += 1;
        }
    }
    Console.WriteLine(prod_name2 + " " + count_A);

    Console.ReadKey();
}
```

14. Solution

```
static void Main() {
    string word;

    Dictionary<string, string> morseAlphabet = new Dictionary<string, string>{
        { "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 a word: ");
    word = Console.ReadLine();

    foreach (var letter in word) {
        Console.Write(morseAlphabet[letter.ToString().ToUpper()]);
    }
    Console.ReadKey();
}
```

Chapter 25

25.4 Review Questions: True/False

1. True
2. True
3. True
4. False
5. False
6. True
7. True
8. True
9. True
10. True
11. False
12. 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. True
21. True
22. True
23. False
24. False

26.12 Review Exercises

1. Solution

```
static double find_max(double a, double b) {
    double maximum;

    if (a > b) {
        maximum = a;
    }
    else {
        maximum = b;
    }
    return maximum;
}
```

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

```
static double find_sum(double a, double b, double c) {
    return a + b + c;
}
```

4. Solution

```
static double find_avg(double a, double b, double c, double d) {
    return (a + b + c + d) / 4;
}
```

5. Solution

```
static void display_max(double a, double b, double c) {
    double maximum;

    maximum = a;
    if (b > maximum) {
        maximum = b;
    }
    if (c > maximum) {
        maximum = c;
    }
    Console.WriteLine(maximum);
}
```

6. Solution

```
static double find_min(double a, double b) {
    double minimum;
    minimum = a;
    if (b < minimum) {
        minimum = b;
    }
    return minimum;
}

static void Main() {
    double x1, x2, x3, x4, temp1, temp2;

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

    //Or as follows (2nd approach)
    Console.WriteLine(find_min(find_min(x1, x2), find_min(x3, x4)));
    Console.ReadKey();
}
```

7. Solution

```
static bool get_input() {
    string answer;
    Console.Write("Enter Yes or No: ");
    answer = Console.ReadLine();
    if (answer.ToUpper() == "YES") {
        return true;
    }
    else {
        return false;
    }
}

static double find_area(double b, double h) {
    return b * h;
}

static void Main() {
    double bas, height;
    bool answer;

    do {
        Console.Write("Enter the base of the parallelogram: ");
        bas = Double.Parse(Console.ReadLine());

        Console.Write("Enter the height of the parallelogram: ");
        height = Double.Parse(Console.ReadLine());

        Console.WriteLine("Area = " + find_area(bas, height));

        Console.WriteLine("Would you like to repeat? ");
        answer = get_input();    //Or you can write...
    } while (answer == true);    //while(get_input() == true);
    Console.ReadKey();
}
```

Chapter 27

27.2 Review Exercises

1. Solution

```
static double Kelvin_to_Fahrenheit(double kelvin) {
    return 1.8 * kelvin - 459.67;
}

static double Kelvin_to_Celsius(double kelvin) {
    return kelvin - 273.15;
}

static void Main() {
    double k;

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

    Console.WriteLine("Fahrenheit: " + Kelvin_to_Fahrenheit(k));
    Console.Write("Celsius: " + Kelvin_to_Celsius(k));
    Console.ReadKey();
}
```

2. Solution

```
static int num_of_days(int month) {
    int days;

    if (month == 4 || month == 6 || month == 9 || month == 11) {
        days = 30;
    }
    else if (month == 2) {
        days = 28;
    }
    else {
        days = 31;
    }
    return days;
}

static void Main() {
    int i, x, y, total;

    Console.Write("Enter a month: ");
    x = Int32.Parse(Console.ReadLine());

    Console.Write("Enter a second month: ");
    y = Int32.Parse(Console.ReadLine());

    total = 0;
    for (i = x; i <= y; i++) {
        total += num_of_days(i);
    }

    Console.WriteLine(total);
    Console.ReadKey();
}
```

3. Solution

```
static int dice() {
    Random rnd = new Random(Guid.NewGuid().GetHashCode());
    return rnd.Next(1, 7);
}

static void Main() {
    int i, dice1, dice2, player;

    string[] names = new string[2];
    Console.Write("Player 1 enter name: ");
    names[0] = Console.ReadLine();
    Console.Write("Player 2 enter name: ");
    names[1] = Console.ReadLine();

    int[] total = new int[2];

    for (player = 0; player <= 1; player++) {
        total[player] = 0;
        for (i = 1; i <= 10; i++) {
            Console.Write(names[player]);
            Console.WriteLine(", hit any key to roll the dice!");
            Console.ReadKey();

            dice1 = dice();
            dice2 = dice();
            Console.WriteLine(dice1 + " " + dice2);
            total[player] += dice1 + dice2;
        }
    }

    if (total[0] == total[1]) {
        Console.WriteLine("Tie!");
    }
    else if (total[0] > total[1]) {
        Console.WriteLine(names[0] + " wins!");
    }
    else {
        Console.WriteLine(names[1] + " wins!");
    }
    Console.ReadKey();
}
```

4. Solution

```
static void bmi(double w, double h) {
    double b;

    b = w * 703 / Math.Pow(h, 2);
    if (b < 16) {
        Console.WriteLine("You must add weight.");
    }
    else if (b < 18.5) {
        Console.WriteLine("You should add some weight.");
    }
    else if (b < 25) {
        Console.WriteLine("Maintain your weight.");
    }
    else if (b < 30) {
        Console.WriteLine("You should lose some weight.");
    }
    else {
        Console.WriteLine("You must lose weight.");
    }
}

static void Main() {
    double height, weight;
    int age;

    Console.Write("Enter your weight (in pounds): ");
    weight = Double.Parse(Console.ReadLine());

    Console.WriteLine("Enter your age: ");
    age = Int32.Parse(Console.ReadLine());

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

    if (age < 18) {
        Console.WriteLine("I can't calculate your BMI. You must be adult!");
    }
    else {
        bmi(weight, height);
    }

    Console.ReadKey();
}
```

5. *Solution*

```
const int CARS = 40;
const int GAS = 1;
const int DIESEL = 2;
const int HYBRID = 3;

static int get_choice() {
    int choice;
    Console.WriteLine("1. Gas");
    Console.WriteLine("2. Diesel");
    Console.WriteLine("3. Hybrid");
    Console.Write("Enter type of the car: ");
    choice = Int32.Parse(Console.ReadLine());
    return choice;
}

static int get_days() {
    int days;

    Console.Write("Enter total number of rental days: ");
    days = Int32.Parse(Console.ReadLine());
    return days;
}

static double get_charge(int car_type, int rental_days) {
    double charge;

    if (car_type == GAS) {
        if (rental_days <= 5) {
            charge = rental_days * 24;
        }
        else {
            charge = rental_days * 22;
        }
    }
    else if (car_type == DIESEL) {
        if (rental_days <= 5) {
            charge = rental_days * 28;
        }
        else {
            charge = rental_days * 25;
        }
    }
    else {
        if (rental_days <= 5) {
            charge = rental_days * 30;
        }
        else {
            charge = rental_days * 28;
        }
    }
    return charge;
}

static void Main() {
    int i, count;
    double charge, total;

    int[] rented_car_types = new int[CARS];
    int[] rented_days = new int[CARS];

    for (i = 0; i <= CARS - 1; i++) {
        rented_car_types[i] = get_choice();
        rented_days[i] = get_days();
    }

    total = 0;
    for (i = 0; i <= CARS - 1; i++) {
        charge = get_charge(rented_car_types[i], rented_days[i]);
        Console.WriteLine("Amount to pay, car No " + (i + 1) + ": " + charge);
        total += charge;
    }

    count = 0;
    for (i = 0; i <= CARS - 1; i++) {
        if (rented_car_types[i] == HYBRID) {
            count += 1;
        }
    }
    Console.WriteLine("Hybrids rented: " + count);

    Console.WriteLine("Total profit: " + total);
    Console.ReadKey();
}
```

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. True
11. True
12. False
13. True
14. False
15. False

28.9 Review Exercises

1. Solution

```
class Trigonometry {
    public double square_area(double side) {
        return side * side;
    }

    public double rectangle_area(double bas, double height) {
        return bas * height;
    }

    public double triangle_area(double bas, double height) {
        return bas * height / 2;
    }
}

static void Main() {
    double sqr_side, rctngl_base, rctngl_height, trngl_base, trngl_height;

    Trigonometry tr = new Trigonometry();

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

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

    Console.Write("Enter triangle base: ");
    trngl_base = Double.Parse(Console.ReadLine());
    Console.Write("Enter triangle height: ");
    trngl_height = Double.Parse(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));
    Console.ReadKey();
}
```

2. *Solution*

```
class Pet {
    public string kind;
    public int legs_number;

    public void start_running() {
        Console.WriteLine("Pet is running");
    }

    public void stop_running() {
        Console.WriteLine("Pet stopped");
    }
}

static void Main() {
    Pet pet1 = new Pet();
    pet1.kind = "dog";
    pet1.legs_number = 4;

    Pet pet2 = new Pet();
    pet2.kind = "monkey";
    pet2.legs_number = 2;

    pet1.start_running();
    pet2.start_running();
    pet1.stop_running();
    Console.ReadKey();
}
```

3. *Solution*

```
class Pet {
    private string kind;
    private int legs_number;

    //Define the constructor
    public Pet(string k, int l) {
        //Initialize properties
        Kind = k;
        Legs_number = l;
    }

    //Defines a public property
    public string Kind {
        //Define the getter
        get {
            return kind;
        }
        //Define the setter
        set {
            if (value != "") {
                kind = value;
            }
            else {
                throw new Exception("Cannot be empty");
            }
        }
    }

    //Defines a public property
    public int Legs_number {
        //Define the getter
        get {
            return legs_number;
        }
        //Define the setter
        set {
            if (value >= 0) {
                legs_number = value;
            }
            else {
                throw new Exception("Cannot be negative");
            }
        }
    }

    public void start_running() {
        Console.WriteLine("Pet is running");
    }

    public void stop_running() {
        Console.WriteLine("Pet stopped");
    }
}

static void Main() {
    Pet pet1 = 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

    Console.ReadKey();
}
```

4. *Solution*

```
const int BOXES = 3;

class Box {
    public double width;
    public double length;
    public double height;

    //Define the constructor
    public Box(double w, double l, double h) {
        //Initialize fields
        this.width = w;
        this.length = l;
        this.height = h;
    }

    public void display_volume() {
        Console.WriteLine("Volume: " +
                            (this.width * this.length * this.height));
    }

    public void display_dimensions() {
        Console.WriteLine(this.width + " x " +
                            this.length + " x " + this.height);
    }
}

static void Main() {
    int i;
    double w, l, h;

    Box[] list_of_obj = new Box[BOXES]; //create an array

    for (i = 0; i <= BOXES - 1; i++) {
        Console.Write("Enter width: ");
        w = Double.Parse(Console.ReadLine());
        Console.Write("Enter length: ");
        l = Double.Parse(Console.ReadLine());
        Console.Write("Enter height: ");
        h = Double.Parse(Console.ReadLine());

        //add each new object to the array
        list_of_obj[i] = new Box(w, l, h);
    }

    for (i = 0; i <= BOXES - 1; i++) {
        list_of_obj[i].display_dimensions();
        list_of_obj[i].display_volume();
    }
    Console.ReadKey();
}
```

5. *Solution*

```
const int BOXES = 3;

class Box {
    private double width;
    private double length;
    private double height;

    //Define the constructor
    public Box(double w, double l, double h) {
        //Initialize properties
        Width = w;
        Length = l;
        Height = h;
    }

    //Define public property Width
    public double Width {
        //Define the getter
        get {
            return width;
        }
        //Define the setter
        set {
            if (value > 0) {
                width = value;
            }
            else {
                throw new Exception("Cannot be negative or zero");
            }
        }
    }

    //Define public property Length
    public double Length {
        //Define the getter
        get {
            return length;
        }
        //Define the setter
        set {
            if (value > 0) {
                width = value;
            }
            else {
                throw new Exception("Cannot be negative or zero");
            }
        }
    }

    //Define public property Height
    public double Height {
        //Define the getter
        get {
            return height;
        }
        //Define the setter
        set {
            if (value > 0) {
                width = value;
            }
            else {
                throw new Exception("Cannot be negative or zero");
            }
        }
    }

    public void display_volume() {
        Console.WriteLine("Volume: " +
                            (this.width * this.length * this.height));
    }

    public void display_dimensions() {
        Console.WriteLine(this.width + " x " +
                            this.length + " x " + this.height);
    }
}

static void Main() {
    int i;
    double w, l, h;

    Box[] list_of_obj = new Box[BOXES]; //create an array

    for (i = 0; i <= BOXES - 1; i++) {
        Console.Write("Enter width: ");
        w = Double.Parse(Console.ReadLine());
        Console.Write("Enter length: ");
        l = Double.Parse(Console.ReadLine());
        Console.Write("Enter height: ");
        h = Double.Parse(Console.ReadLine());

        //add each new object to the list
        list_of_obj[i] = new Box(w, l, h);
    }

    for (i = 0; i <= BOXES - 1; i++) {
        list_of_obj[i].display_dimensions();
        list_of_obj[i].display_volume();
    }
    Console.ReadKey();
}
```

6. Solution

```
class Cube {
    public double edge;

    //Define the constructor
    public Cube(double edge) {
        this.edge = edge;   //Initialize field
    }

    public void display_volume() {
        Console.WriteLine("Volume: " + Math.Pow(this.edge, 3));
    }

    public void display_one_surface() {
        Console.WriteLine("One surface: " + Math.Pow(this.edge, 2));
    }

    public void display_total_surface() {
        Console.WriteLine("Total surface: " + 6 * Math.Pow(this.edge, 2));
    }
}

static void Main() {
    double edge;

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

    Cube cube1 = new Cube(edge);

    cube1.display_volume();
    cube1.display_one_surface();
    cube1.display_total_surface();
    Console.ReadKey();
}
```

7. Solution

```
class Cube {
    private double edge;

    //Define the constructor
    public Cube(double edge) {
        Edge = edge;   //Initialize property
    }

    //Define public property Edge
    public double Edge {
        //Define the getter
        get {
            return edge;
        }
        //Define the setter
        set {
            if (value > 0) {
                edge = value;
            }
            else {
                throw new Exception("Cannot be negative or zero");
            }
        }
    }

    public void display_volume() {
        Console.WriteLine("Volume: " + Math.Pow(this.edge, 3));
    }

    public void display_one_surface() {
        Console.WriteLine("One surface: " + Math.Pow(this.edge, 2));
    }

    public void display_total_surface() {
        Console.WriteLine("Total surface: " + (6 * Math.Pow(this.edge, 2)));
    }
}

static void Main() {
    double edge;

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

    Cube cube1 = new Cube(edge);

    cube1.display_volume();
    cube1.display_one_surface();
    cube1.display_total_surface();
    Console.ReadKey();
}
```

8. Solution

```
static void 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");
}

class Circle {
    private double radius = -1;

    //Define public property Radius
    public double Radius {
        //Define the getter
        get {
            if (radius > 0) {
                return radius;
            }
            else {
                throw new Exception("Radius is not set");
            }
        }
        //Define the setter
        set {
            if (value > 0) {
                radius = value;
            }
            else {
                throw new Exception("Cannot be negative or zero");
            }
        }
    }

    public double get_diameter() {
        return 2 * Radius;
    }

    public double get_area() {
        return 3.14 * Math.Pow(Radius, 2);
    }

    public double get_perimeter() {
        return 2 * 3.14 * Radius;
    }
}
```

```
static void Main() {
    int choice;
    double radius;

    Circle circle1 = new Circle();

    do {
        display_menu();
        Console.Write("Enter a choice: ");
        choice = Int32.Parse(Console.ReadLine());

        if (choice == 1) {
            Console.Write("Enter radius: ");
            radius = Double.Parse(Console.ReadLine());
            circle1.Radius = radius;
        }
        else if (choice == 2) {
            Console.WriteLine("Radius: " + circle1.Radius);
        }
        else if (choice == 3) {
            Console.WriteLine("Diameter: " + circle1.get_diameter());
        }
        else if (choice == 4) {
            Console.WriteLine("Area: " + circle1.get_area());
        }
        else if (choice == 5) {
            Console.WriteLine("Perimeter: " + circle1.get_perimeter());
        }
    } while (choice != 6);
}
```

9. Solution

```
class Info {
    private string user_text;

    public string User_text {
        //Define the getter
        get {
            return user_text;
        }
        //Define the setter
        set {
            if (value != "") {
                user_text = value;
            }
            else {
                throw new Exception("Cannot be set to empty");
            }
        }
    }

    public int get_spaces_count() {
        int count = 0;
        foreach (var character in User_text) {
            if (character.ToString() == " ") {
                count += 1;
            }
        }
        return count;
    }

    public int get_words_count() {
        return this.get_spaces_count() + 1;
    }

    public int get_vowels_count() {
        int count = 0;
        foreach (var character in User_text) {
            if ("aeiou".IndexOf(character) > -1) {
                count += 1;
            }
        }
        return count;
    }

    public int get_letters_count() {
        return User_text.Length - this.get_spaces_count();
    }
}

static void Main() {
    Info inf = 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());
    Console.ReadKey();
}
```

10. Solution

```
static void display_menu() {
    Console.WriteLine("1. Encryption/Decryption key");
    Console.WriteLine("2. Encrypt a message");
    Console.WriteLine("3. Decrypt a message");
    Console.WriteLine("4. Exit");
}

class EncryptDecrypt {
    private int encr_decr_key = -1;
    private string alphabet = " abcdefghijklmnopqrstuvwxyz";  //space is
                                                              //a valid character!
    public int Encr_decr_key {
        //Define the getter
        get {
            if (encr_decr_key != -1) {
                return encr_decr_key;
            }
            else {
                throw new Exception("Key is not set");
            }
        }
        //Define the setter
        set {
            if (value >= 1 && value <= 26) {
                encr_decr_key = value;
            }
            else {
                throw new Exception("Must be between 1 and 26");
            }
        }
    }

    public string encrypt(string message) {
        string new_letter, return_value = "";
        int index, new_index;

        foreach (var character in message) {
            index = this.alphabet.IndexOf(character);
            new_index = index + Encr_decr_key;
            if (new_index >= 27) {
                new_index -= 27;
            }
            new_letter = this.alphabet[new_index].ToString();
            return_value += new_letter;
        }
        return return_value;
    }

    public string decrypt(string enc_message) {
        string new_letter, return_value = "";
        int index, new_index;

        foreach (var character in enc_message) {
            index = this.alphabet.IndexOf(character);
            new_index = index - Encr_decr_key;
            if (new_index < 0) {
                new_index += 27;
            }
            new_letter = this.alphabet[new_index].ToString();
            return_value += new_letter;
        }
        return return_value;
    }
}

static void Main() {
    string text;
    int choice;

    EncryptDecrypt ed = new EncryptDecrypt();

    do {
        display_menu();

        Console.Write("Enter a choice: ");
        choice = Int32.Parse(Console.ReadLine());

        if (choice == 1) {
            Console.Write("Enter encryption/decryption key: ");
            ed.Encr_decr_key = Int32.Parse(Console.ReadLine());
        }
        else if (choice == 2) {
            Console.Write("Enter message to encrypt: ");
            text = Console.ReadLine();
            Console.WriteLine("Encrypted message: " + ed.encrypt(text));
        }
        else if (choice == 3) {
            Console.Write("Enter message to decrypt: ");
            text = Console.ReadLine();
            Console.WriteLine("Decrypted message: " + ed.decrypt(text));
        }
    } while (choice != 4);
}
```