Patuakhali Science and Technology University

Assignment on

"Deitel Book Exercise 4.1 to 4.16 and Section 3.2 solve"

Course Code: CCE-121

Course Title: Object Oriented Programming

Level - I; Semester - II

Submitted By

Name: M.D. Sakibul Islam Shovon
ID: 2302056
REG: 11834
Session: 2023-2024
Faculty of Computer Science and Engineering

Submitted To

Prof. Dr. Md. Samsuzzaman
Professor of Computer and Communication Engineering Department
Faculty of Computer Science and Engineering

Chapter's Exercise Questions

4.1 Fill in the blanks in each of the following statements:

a) All programs can be written in terms of three types of control structures: **sequence**, **selection** and **repetition**.

b) The **if…else** statement is used to execute one action when a condition is true and another when that condition is false.

c) Repeating a set of instructions a specific number of times is called **counter-controlled (or definite)** repetition.

d) When it's not known in advance how many times a set of statements will be repeated, a(n) **sentinel, signal, flag or dummy** value can be used to terminate the repetition.

e) The **sequence** structure is built into Java; by default, statements execute in the order they appear.

f) Instance variables of types char, byte, short, int, long, float and double are all given the value **0(zero)** by default.

g) Java is a(n) **strongly typed** language; it requires all variables to have a type.

h) If the increment operator is **prefixed** to a variable, first the variable is incremented by 1, then its new value is used in the expression.

4.2 State whether each of the following is true or false. If false, explain why.

a) An algorithm is a procedure for solving a problem in terms of the actions to execute and the order in which they execute. **True**.

b) A set of statements contained within a pair of parentheses is called a block. **False**.

A set of statements contained within a pair of braces ({ and }) is called a block.

c) A selection statement specifies that an action is to be repeated while some condition remains true. **False**.

A repetition statement specifies that an action is to be repeated while some condition remains true.

d) A nested control statement appears in the body of another control statement. **True**.

e) Java provides the arithmetic compound assignment operators +=, -=, *=, /= and %= for abbreviating assignment expressions. **True**.

f) The primitive types (boolean, char, byte, short, int, long, float and double) are portable across only Windows platforms. **False**.

The primitive types (boolean, char, byte, short, int, long, float and double) are portable across all computer platforms that support Java.

g) Specifying the order in which statements execute in a program is called program control. **True**.

h) The unary cast operator (double) creates a temporary integer copy of its operand. **False**.

The unary cast operator (double) creates a temporary floating-point copy of its operand.

i) Instance variables of type boolean are given the value true by default. **False**. Instance variables of type boolean are given the value false by default.

j) Pseudocode helps you think out a program before attempting to write it in a programming language. **True**.

4.3 Write four different Java statements that each add 1 to integer variable x.

Ans: x = x + 1;

x += 1;

++x;

x++;

4.4 Write Java statements to accomplish each of the following tasks:

a) Use one statement to assign the sum of x and y to z, then increment x by 1.

b) Test whether variable count is greater than 10. If it is, print "Count is greater than 10".

c) Use one statement to decrement the variable x by 1, then subtract it from variable total

and store the result in variable total.

d) Calculate the remainder after q is divided by divisor, and assign the result to q. Write

this statement in two different ways.

Ans: a) z = x++ + y;

b) if (count > 10)

System.out.println("Count is greater than 10");

c) total -= --x;

d) q %= divisor;

q = q % divisor;

4.5 Write a Java statement to accomplish each of the following tasks:

a) Declare variables sum of type int and initialize it to 0.

b) Declare variables x of type int and initialize it to 1.

c) Add variable x to variable sum, and assign the result to variable sum.

d) Print "The sum is: ", followed by the value of variable sum.

Ans: a) int sum = 0;

b) int x = 1;

c) sum += x; or sum = sum + x;

d) System.out.printf("The sum is: %d%n", sum);

4.6 Combine the statements that you wrote in Exercise 4.5 into a Java application that calculates and prints the sum of the integers from 1 to 10. Use a while statement to loop through the calculation and increment statements. The loop should terminate when the value of x becomes 11.

Ans:

```
public class Calculate
{
   public static void main(String[] args)
   {
      int sum = 0;
      int x = 1;
      while (x <= 10)
      {
         sum += x;
         ++x;
      }
      System.out.printf("The sum is: %d%n", sum);
   }
}
```

4.7 Determine the value of the variables in the statement product *= x++; after the calculation

is performed. Assume that all variables are type int and initially have the value 5.

Ans:

product = 25, x=6

4.8 Identify and correct the errors in each of the following sets of code:

a) while (c <= 5)

{

product *= c;

++c;

b) if (gender == 1)

System.out.println("Woman");

else;

System.out.println("Man");

Ans:

a) Error: The closing right brace of the while statement's body is missing.

Correction: Add a closing right brace after the statement ++c;.

b) Error: The semicolon after else results in a logic error. The second output statement

will always be executed.

Correction: Remove the semicolon after else.

4.9 What is wrong with the following while statement?

```
while (z >= 0)
 sum += z;
```

The value of the variable z is never changed in the while statement. Therefore, if the loopcontinuation condition (z >= 0) is true, an infinite loop is created. To prevent an infinite loop from occurring, z must be decremented so that it eventually becomes less than 0.

4.10 Compare and contrast the if single-selection statement and the while repetition statement. How are these two statements similar? How are they different?

Ans:

the if single-selection statement is used for making decisions and executing a block of code once based on a condition, while the while repetition statement is used for executing a block of code multiple times as long as a condition remains true.

4.11 Explain what happens when a Java program attempts to divide one integer by another. What happens to the fractional part of the calculation? How can you avoid that outcome?

Ans:

Dividing two integers results in integer division—any fractional part of the calculation is lost (i.e., truncated). For example, 7 ÷ 4, which yields 1.75 in conventional arithmetic, truncates to 1 in integer arithmetic, rather than rounding to 2. To obtain a floating-point result from dividing integer values, a programmer must temporarily treat these values as floating-point numbers in the calculation by using the unary cast operator (double). As long as the (double) cast operator is applied to any variable in the calculation, the calculation will yield a double result which can be assigned to a double variable.

4.12 Describe the two ways in which control statements can be combined.

Ans:

Control statements can be “attached” (that is, stacked) to one another by connecting the exit point of one to the entry point of the next. Control statements also may be nested by placing one inside another.

4.13 What type of repetition would be appropriate for obtaining an input from the user until the user indicates there is no more input to provide? What type would be appropriate for calculating the factorial of 5? Briefly describe how each of these tasks could be performed.

Ans:

Counter-controlled repetition would be appropriate for calculating the sum of the first 100 positive integers because the number of repetitions is known in advance. The program that performs this task could use a while repetition statement with a counter variable that takes on the values 1 to 100. The program could then add the current counter value to the total variable in each repetition of the loop. Sentinelcontrolled repetition would be appropriate for calculating the sum of an arbitrary number of positive integers. The program that performs this task could use a sentinel value of –1 to mark the end of data entry. The program could use a while repetition statement that totals positive integers from the user until the user enters the sentinel value.

4.14 If integers x and y are set to 7 and 3, what is the value of x after x = y++ and x = ++y?

Ans:

Preincrementing a variable causes it to be incremented by 1, and then the new value of the variable is used in the expression in which it appears. Postincrementing a variable causes the current value of the variable to be used in the expression in which it appears, and then the variable’s value is incremented by

1. Preincrementing and postincrementing a variable have the same effect when the incrementing operation appears in a statement by itself.

4.15 Identify and correct the errors in each of the following pieces of code. [Note: There may be more than one error in each piece of code.]

a) if (age >= 65);

```
 System.out.println("Age is greater than or equal to 65");
else
 System.out.println("Age is less than 65)";
```

Ans :

The semicolon at the end of the if condition should be removed. The closing double quote of the second System.out.println should be inside the closing parenthesis.

b) int x == 1, total == 0;

```
while (x <= 10)
{
 total ++x;
 System.out.println(x);
}
```

Ans:

The variable total should be initialized to zero.

c) while (x <= 100)

```
 total += x;
 ++x;
```

Ans:

The two statements should be enclosed in braces to properly group them into the body of the while; otherwise the loop will be an infinite loop.

d) while (y =! 0)

{

System.out.println (y);

Ans:

The ++ operator should be changed to --; otherwise the loop will be an infinite loop if y starts with a value greater than 0. The closing brace for the while loop is missing.

4.16 What does the following program print

```
public class Mystery {
   public static void main(String[] args) {
      int y;
      int x = 1;
      int total = 0;

      while (x <= 10) {
         y = x * x;
         System.out.println(y);
         total += y;
         ++x;
      }
```

```
        System.out.printf("Total is %d\n", total);
    }
}
```

Ans:

```
1
4
9
16
25
36
49
64
81
100
Total is 385
```

Given Questions

Section 3.2

▼3.2.1

List six relational operators.

Answer:

- < : less than
- <= : less than or equal to
- > : greater than
- >= : greater than or equal to
- == : equal to
- /= : not equal to

▼3.2.2

Assuming that x is 1, show the result of the following Boolean expressions:x > 0

x < 0

x != 0

x >= 0

x != 1

Answer:

true

false

true

true

false

▼3.2.3

Can the following conversions involving casting be allowed? Write a test program to verify it.boolean b = true;
i = (int)b;
int i = 1;
boolean b = (boolean)i;

Answer:

No. Boolean values cannot be cast to other types.

Section 3.3

▼3.3.1

Write an if statement that assigns 1 to x if y is greater than 0.

Answer:

```
if (y > 0)
  x = 1;
```

▼3.3.2

Write an if statement that increases pay by 3% if score is greater than 90.

Answer:

```
if (y > 0)
  x = 1;
```

▼3.3.3

What is wrong in the following code?if radius >= 0
{

```
area = radius * radius * PI;
System.out.println("The area for the circle of " +
" radius " + radius + " is " + area);
}
```

Answer:

The parentheses is reuqired for the conidition radius >= 0.

Section 3.4

▼3.4.1

Write an if statement that increases pay by 3% if score is greater than 90, otherwise increases pay by 1%.

Answer:

```
if (score > 90)
  pay *= 1.03;
else
  pay *= 1.01;
```

▼3.4.2

What is the output of the code in (a) and (b) if number is 30? What if number is 35?(a)

```
if (number % 2 == 0)
System.out.println(number + " is even.");

System.out.println(number + " is odd.");
```

(b)

```
if (number % 2 == 0)
System.out.println(number + " is even.");
```

else
System.out.println(number + " is odd.");

Answer:

If number is 30, (a) displays

30 is even

30 is odd

(b) displays

30 is even

If number is 35, (a) displays

35 is odd

(b) displays

35 is odd

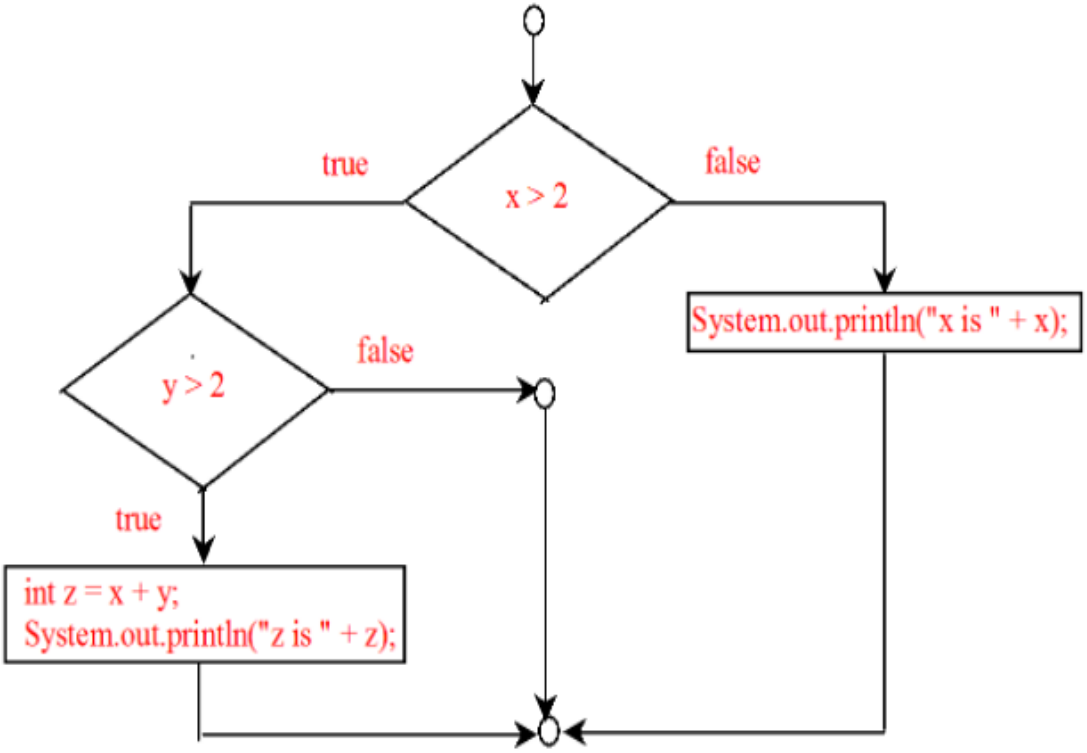
Section 3.5

▼3.5.1

Suppose x = 3 and y = 2; show the output, if any, of the following code. What is the output if x = 3 and y = 4? What is the output if x = 2 and y = 2? Draw a flowchart of the code.if (x > 2) {
if (y > 2) {
z = x + y;
System.out.println("z is " + z);
}
}
else

System.out.println("x is " + x);

Answer:



▼3.5.2

Suppose x = 2 and y = 3. Show the output, if any, of the following code. What is the output if x = 3 and y = 2? What is the output if x = 3 and y = 3?if (x > 2)

if (y > 2) {

int z = x + y;

System.out.println("z is " + z);

}

else

System.out.println("x is " + x);

Answer:

```
if (x > 2)
  if (y > 2) {
    int z = x + y;
    System.out.println("z is " + z);
  }
  else
    System.out.println("x is " + x);
```

and same as

```
if (x > 2) {
  if (y > 2) {
    int z = x + y;
    System.out.println("z is " + z);
  }
  else
    System.out.println("x is " + x);
}
```

No output if x = 2 and y = 3. Output is "x is 3" if x = 3 and y = 2. Output is "z is 6" if x = 3 and y = 3.

▼3.5.3

What is wrong in the following code?if (score >= 60.0)
System.out.println("D");
else if (score >= 70.0)
System.out.println("C");
else if (score >= 80.0)
System.out.println("B");

```
else if (score >= 90.0)
System.out.println("A");
else
System.out.println("F");
```

Answer:

Consider score 90, what will be the grade? It will be D.

Section 3.6

▼3.6.1

Which of the following statements are equivalent? Which ones are correctly indented?(a)

```
if (i > 0) if
(j > 0)
x = 0; else
if (k > 0) y = 0;
else z = 0;
```

(b)

```
if (i > 0) {
if (j > 0)
x = 0;
else if (k > 0)
y = 0;
}
else
z = 0;
```

(c)

```
if (i > 0)
if (j > 0)
x = 0;
else if (k > 0)
y = 0;
else
z = 0;
```

(d)

```
if (i > 0)
if (j > 0)
x = 0;
else if (k > 0)
y = 0;
else
z = 0;
```

Answer:

a, c, and d are the same. (B) and (C) are correctly indented.

▼3.6.2

Rewrite the following statement using a Boolean expression:

```
if (count % 10 == 0)
newLine = true;
else
newLine = false;
```

Answer:

```
newLine = (count % 10 == 0);
```

▼3.6.3

Are the following statements correct? Which one is better?(a)

```
if (age < 16)
System.out.println
("Cannot get a driver's license");
if (age >= 16)
System.out.println
("Can get a driver's license");
```

(b)

```
if (age < 16)
System.out.println
("Cannot get a driver's license");
else
System.out.println
("Can get a driver's license");
```

Answer:

Both are correct. (b) is better.

▼3.6.4

What is the output of the following code if number is 14, 15, or 30?(a)

```
if (number % 2 == 0)
System.out.println
(number + " is even");
if (number % 5 == 0)
System.out.println
(number + " is multiple of 5");
```

(b)

```
if (number % 2 == 0)
System.out.println
(number + " is even");
else if (number % 5 == 0)
System.out.println
(number + " is multiple of 5");
```

Answer:

For (a) if number is 14, the output is

14 is even

if number is 15, the output is

15 is multiple of 5

if number is 30, the output is

30 is even

30 is multiple of 5

For (b) if number is 14, the output is

14 is even

If number is 15, the output is

15 is multiple of 5

if number is 30, the output is

30 is even

Section 3.7

▼3.7.1

Which of the following is a possible output from invoking

Math.random()?323.4, 0.5, 34, 1.0, 0.0, 0.234

Answer:

0.5, 0.0, 0.234

▼3.7.2

a. How do you generate a random integer i such that 0 <= i < 20 ?
b. How do you generate a random integer i such that 10 <= i < 20
c. How do you generate a random integer i such that 10 <= i <= 50
d. Write an expression that returns 0 or 1 randomly.

Answer:

(a) (int)(Math.random() * 20)
(b) 10 + (int)(Math.random() * 10)
(c) 10 + (int)(Math.random() * 41)
(d) (int)(Math.random() * 2)

Section 3.9

▼3.9.1

Are the following two statements equivalent?

(a)

```
if (income <= 10000)
tax = income * 0.1;
else if (income <= 20000)
tax = 1000 +
(income - 10000) * 0.15;
```

(b)

```
if (income <= 10000)
tax = income * 0.1;
else if (income > 10000 &&
```

income <= 20000)
tax = 1000 +
(income - 10000) * 0.15;

Answer:

Yes

Section 3.10

▼3.10.1

Assuming that x is 1, show the result of the following Boolean expressions.(true) && (3 > 4)
!(x > 0) && (x > 0)
(x > 0) || (x < 0)
(x != 0) || (x == 0)
(x >= 0) || (x < 0)
(x != 1) == !(x == 1)

Answer:

(true) && (3 > 4) is false
!(x > 0) && (x > 0) is false
(x > 0) || (x < 0) is true
(x != 0) || (x == 0) is true
(x >= 0) || (x < 0) is true
(x != 1) == !(x == 1) is true

▼3.10.2

(a) Write a Boolean expression that evaluates to true if a number stored in variable num is between 1 and 100.
(b) Write a Boolean expression that evaluates to true if a number stored

in variable num is between 1 and 100 or the number is negative.

Answer:

(a) (num > 1) && (num < 100)
(b) (num > 1) && (num < 100) || num < 0

▼3.10.3

(a) Write a Boolean expression for |x - 5| < 4.5.

(b) Write a Boolean expression for |x - 5| > 4.5.

Answer:

(a) (x - 5) < 4.5 && (x - 5) > -4.5
(b) (x - 5) > 4.5 || (x - 5) < -4.5

▼3.10.4

Assume that x and y are int type. Which of the following are legal Java expressions?x > y > 0

x = y && y

x /= y

x or y

x and y

(x != 0) || (x = 0)

Answer:

x > y > 0 is incorrect
x = y && y is incorrect
x /= y is correct
x or y is incorrect
x and y is incorrect
(x != 0) || (x = 0) is incorrect on x = 0. It should be x == 0.

▼3.10.5

Are the following two expressions the same?

(a) x % 2 == 0 && x % 3 == 0
(b) x % 6 == 0

Answer:

Yes.

▼3.10.6

What is the value of the expression x >= 50 && x <= 100 if x is 45, 67, or 101?

Answer:

If x is 45, the expression is false.
If x is 67, the expression is true.
If x is 101, the expression is false.

▼3.10.7

Suppose, when you run the following program, you enter the input 2 3 6 from the console. What is the output?public class Test {
public static void main(String[] args) {
java.util.Scanner input = new java.util.Scanner(System.in);
double x = input.nextDouble();
double y = input.nextDouble();
double z = input.nextDouble();

System.out.println("(x < y && y < z) is " + (x < y && y < z));
System.out.println("(x < y || y < z) is " + (x < y || y < z));
System.out.println("!(x < y) is " + !(x < y));
System.out.println("(x + y < z) is " + (x + y < z));
System.out.println("(x + y > z) is " + (x + y > z));
}
}

Answer:

(x < y && y < z) is true
(x < y || y < z) is true
!(x < y) is false
(x + y < z) is true
(x + y > z) is false

▼3.10.8

Write a Boolean expression that evaluates to true if age is greater than 13 and less than 18.

Answer:

age > 13 && age < 18

▼3.10.9

Write a Boolean expression that evaluates to true if weight is greater than 50 pounds or height is greater than 60 inches.

Answer:

weight > 50 || height > 60.

▼3.10.10

Write a Boolean expression that evaluates to true if weight is greater than 50 pounds and height is greater than 60 inches.

Answer:

weight > 50 && height > 60.

▼3.10.11

Write a Boolean expression that evaluates to true if either weight is greater than 50 pounds or height is greater than 60 inches, but not both.

Answer:

weight > 50 ^ height > 60.

Section 3.11

▼3.11.1

How many days in the February of a leap year? Which of the following is a leap year? 500, 1000, 2000, 2016, and 2020?

Answer:

29 days. 500, 1000 are not leap years. 2000, 2016, and 2020 are leap years.

Section 3.12

▼3.12.1

What happens if you enter an integer as 05?

Answer:

It will be the same as entering 5.

Section 3.13

▼3.13.1

What data types are required for a switch variable? If the keyword break is not used after a case is processed, what is the next statement to be executed? Can you convert a switch statement to an equivalent if statement, or vice versa? What are the advantages of using a switch statement?

Answer:

Switch variables must be of char, byte, short, int, or String types. If a break statement is not used, the next case statement is performed. You can always convert a switch statement to an equivalent if statement, but not an if statement to a switch statement. The use of the switch statement can improve readability of the program in some cases. The compiled code for the switch statement is also more efficient than its corresponding if statement.

▼3.13.2

What is y after the following switch statement is executed? Rewrite the code using an if-else statement.x = 3; y = 3;

```
switch (x + 3) {
case 6: y = 1;
default: y += 1;
}
```

Answer:

y is 2.

```
x = 3; y = 3;
if (x + 3 == 6) {
  y = 1;
}
y += 1;
```

▼3.13.3

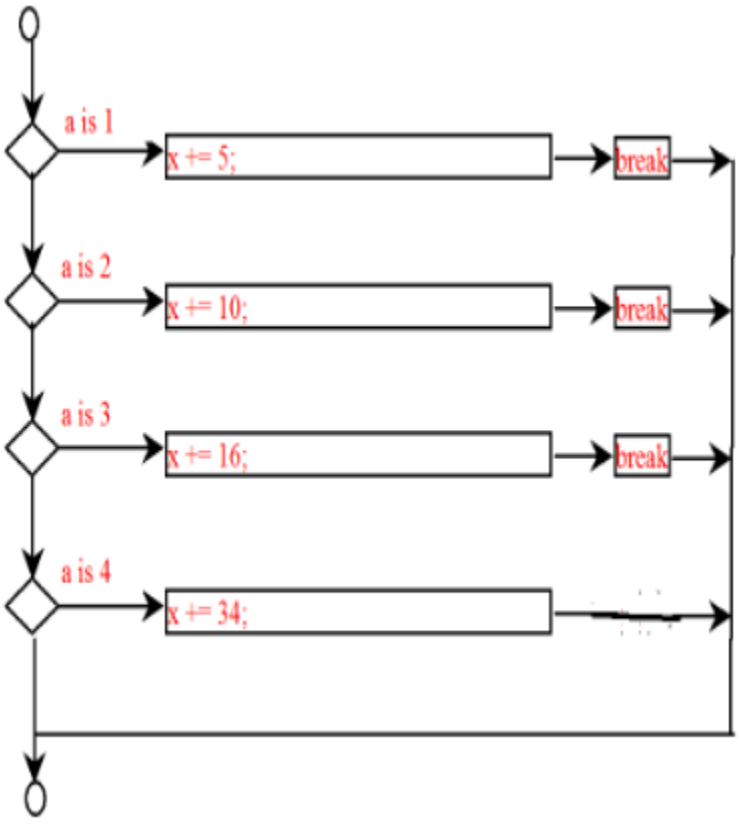
What is x after the following if-else statement is executed? Use a switch statement to rewrite it and draw the flowchart for the new switch statement.int x = 1, a = 3;

```
if (a == 1)
x += 5;
else if (a == 2)
x += 10;
else if (a == 3)
x += 16;
else if (a == 4)
x += 34;
```

Answer:

x is 17

```
switch (a) {
  case 1: x += 5; break;
  case 2: x += 10; break;
  case 3: x += 16; break;
  case 4: x += 34;
}
```



▼3.13.4

Write a switch statement that displays Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, if day is 0, 1, 2, 3, 4, 5, 6,

accordingly.

Answer:

```
switch (day) {
  case 0: System.out.println("Sunday"); break;
  case 1: System.out.println("Monday"); break;
  case 2: System.out.println("Tuesday"); break;
  case 3: System.out.println("Wednesday"); break;
  case 4: System.out.println("Thurday"); break;
  case 5: System.out.println("Friday"); break;
  case 6: System.out.println("Saturday"); break;
}
```

▼3.13.5

Rewrite the switch statement in Listing 3.8 using an if-else statement.

Answer:

```
int remainder = year % 12;
    if (remainder == 0)
      System.out.println("monkey");
    else if (remainder == 1)
      System.out.println("rooster");
    else if (remainder == 2)
      System.out.println("dog");
    else if (remainder == 3)
      System.out.println("pig");
    else if (remainder == 4)
```

```
    System.out.println("rat");
  else if (remainder == 5)
    System.out.println("ox");
  else if (remainder == 6)
    System.out.println("tiger");
  else if (remainder == 7)
    System.out.println("rabbit");
  else if (remainder == 8)
    System.out.println("dragon");
  else if (remainder == 9)
    System.out.println("snake");
  else if (remainder == 10)
    System.out.println("horse");
  else
    System.out.println("sheep");
```

Section 3.14

▼3.14.1

Suppose that, when you run the following program, you enter the input 2 3 6 from the console. What is the output?public class Test {
public static void main(String[] args) {
java.util.Scanner input = new java.util.Scanner(System.in);
double x = input.nextDouble();
double y = input.nextDouble();
double z = input.nextDouble();

```
System.out.println((x < y && y < z) ?
"sorted" : "not sorted");
}
}
```

Answer:

Sorted

▼3.14.2

Rewrite the following if statements using the conditional operator.if

```
(ages >= 16)
ticketPrice = 20;
else
ticketPrice = 10;
```

Answer:

```
ticketPrice = (ages >= 16) ? 20 : 10;
```

▼3.14.3

Rewrite the following conditional expressions using if-else statements.

a. score = (x > 10) ? 3 * scale : 4 * scale;

b. tax = (income > 10000) ? income * 0.2 : income * 0.17 + 1000;

c. System.out.println((number % 3 == 0) ? i : j);

Answer:

(a)

```
if (x > 10)
  score = 3 * scale;
else
```

```
  score = 4 * scale;
```

(b)

```
if (income > 10000)
  tax = income * 0.2;
else
  tax = income * 0.17 + 1000;
```

(c)

```
if (number % 3 == 0)
    System.out.println(i);
else
    System.out.println(j);
```

▼3.14.4

Write conditional expression that returns -1 or 1 randomly.

Answer:

```
(int)(Math.random() * 2) == 0 ? -1 : 1;
```

Section 3.15

▼3.15.1

List the precedence order of the Boolean operators. Evaluate the following expressions:true || true && false
true && true || false

Answer:

The precedence order for boolean operators is !, ^, &&, and ||
true || true && false is true
true && true || false is true

▼3.15.2

True or false? All the binary operators except = are left associative.
Answer:

True.

▼3.15.3

Evaluate the following expressions:2 * 2 - 3 > 2 && 4 - 2 > 5
2 * 2 - 3 > 2 || 4 - 2 > 5

Answer:

Both are false.

▼3.15.4

Is x > 0 && x < 10 the same as x > 0 && x < 10?
Is x > 0 || x < 10 the same as x > 0 || x < 10?
Is (x > 0 || x < 10) && y < 0 the same as (x > 0 || (x < 10 && y < 0))?
Answer:

- x > 0 && x < 10 **is the same** as x > 0 && x < 10
- x > 0 || x < 10 **is the same** as x > 0 || x < 10
- (x > 0 || x < 10) && y < 0 **is NOT the same** as x > 0 || (x < 10 && y < 0)
  Because **operator precedence and grouping changes the meaning**.