

## Answers to Review Questions

### Chapter 3

#### Multiple Choice and True/False

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

#### Find the Error

1. Each `if` clause is prematurely terminated by a semicolon.
2. The `=` operator should be `==`.
3. The conditionally-executed blocks of code should be enclosed in braces.
4. The `case` expressions cannot have relational operators. They must be integer expressions.
5. The `!` operator is only applied to the variable `x`, not the expression. The code should read:

```
if (!(x > 20))
```

6. The statement should use the `&&` operator instead of the `||` operator.
7. The statement should use the `||` operator instead of the `&&` operator.
8. The `:` and `?` are reversed in their positions. The statement should read:  
`z = (a < 10) ? 0 : 7;`
9. The `equalsIgnoreCase` method should be used instead of the `equals` method.

10. The statement is missing the new keyword. It should read:  
`DecimalFormat fmt = new DecimalFormat("00.00");`

### Algorithm Workbench

1. 

```
if (y == 0)
    x = 100;
```
2. 

```
if (y == 10)
    x = 0;
else
    x = 1;
```
3. 

```
if (sales < 10000)
    commission = .10;
else if (sales <= 15000)
    commission = .15;
else
    commission = .20;
```
4. 

```
if (minimum)
    hours = 10;
```
5. 

```
if (amount1 > 10)
{
    if (amount2 < 100)
    {
        if (amount1 > amount2)
        {
            System.out.println(amount1);
        }
        else
        {
            System.out.println(amount2);
        }
    }
}
```
6. 

```
if (grade >= 0 && grade <= 100)
    System.out.println("The number is valid.");
```
7. 

```
if (temperature >= -50 && temperature <= 150)
    System.out.println("The number is valid.");
```
8. 

```
if (hours < 0 || hours > 80)
    System.out.println("The number is not valid.");
```

- ```

9.  if (title1.compareTo(title2) < 0)
        System.out.println(title1 + " " + title2);
    else
        System.out.println(title2 + " " + title1);

10. switch (choice)
    {
        case 1:
            System.out.println("You selected 1.");
            break;
        case 2:
        case 3:
            System.out.println("You selected 2 or 3.");
            break;
        case 4:
            System.out.println("You selected 4.");
            break;
        default:
            System.out.println("Select again please.");
    }

11.  C, A, B
12.  "00000.000"
13.  "0.00"
14.  "000,000,000.00"

```

### Short Answer

1. Conditionally executed code is executed only under a condition, such as an expression being true.
2. If you inadvertently terminate an `if` statement with a semicolon, the compiler will assume that you are placing a null statement there. The null statement, which is an empty statement that does nothing, will become the conditionally executed statement. The statement that you intended to be conditionally executed will be disconnected from the `if` statement and will always execute.
3. By indenting the conditionally executed statements, you are causing them to stand out visually. This is so you can tell at a glance what part of the program the `if` statement executes.
4. The memory addresses of the two `String` objects are compared.
5. A flag is a `boolean` variable that signals when some condition exists in the program. When the flag variable is set to `false`, it indicates the condition does not yet exist. When the flag variable is set to `true`, it means the condition does exist.
6. There is no default action that takes place when none of the `if` clauses' `boolean` expressions are true.
7. It takes two `boolean` expressions as operands and creates a `boolean` expression that is true only when both subexpressions are true.

8. It takes two `boolean` expressions as operands and creates a boolean expression that is true when either of the subexpressions are true.
9. They determine whether a specific relationship exists between two values. The relationships are greater-than, less-than, equal-to, not equal-to, greater-than or equal-to, and less-than or equal-to.
10. A constructor executes automatically when an object is created. Its purpose is to initialize the object's attributes with data and perform any necessary startup operations.