

TEST BANK



SAVITCH



- Classes and Methods
- References
- Scanner, Class Input
- Automatic Boxing
- Enhanced for Loop
- Interfaces
- Inheritance
- Polymorphism
- Encapsulation
- Generics
- Streams and File I/O
- Exception Handling
- ArrayList
- Linked Lists
- Swing GUIs
- Threads

ABSOLUTE
JAVA 3rd Edition

Chapter 3

Flow of Control

■ Multiple Choice

- 1) An if selection statement executes if and only if:
- (a) the Boolean condition evaluates to false.
 - (b) the Boolean condition evaluates to true.
 - (c) the Boolean condition is short-circuited.
 - (d) none of the above.

Answer: B

- 2) A compound statement is enclosed between:
- (a) []
 - (b) { }
 - (c) ()
 - (d) < >

Answer: B

- 3) A multi-way if-else statement
- (a) allows you to choose one course of action.
 - (b) always executes the else statement.
 - (c) allows you to choose among alternative courses of action.
 - (d) executes all Boolean conditions that evaluate to true.

Answer: C

- 4) The controlling expression for a switch statement includes all of the following types except:
- (a) char
 - (b) int
 - (c) byte
 - (d) double

Answer: D

- 5) To compare two strings lexicographically the String method _____ should be used.
- (a) equals
 - (b) equalsIgnoreCase
 - (c) compareTo
 - (d) ==

Answer: C

- 6) When using a compound Boolean expression joined by an && (AND) in an if statement:
- (a) Both expressions must evaluate to true for the statement to execute.
 - (b) The first expression must evaluate to true and the second expression must evaluate to false for the statement to execute.
 - (c) The first expression must evaluate to false and the second expression must evaluate to true for the statement to execute.
 - (d) Both expressions must evaluate to false for the statement to execute.

Answer: A

- 7) The OR operator in Java is represented by:
- (a) !
 - (b) &&
 - (c) ||
 - (d) None of the above

Answer: C

- 8) The negation operator in Java is represented by:
- (a) !
 - (b) &&
 - (c) ||
 - (d) None of the above

Answer: A

- 9) The _____ operator has the highest precedence.
- (a) *
 - (b) dot
 - (c) +=
 - (d) decrement

Answer: B

- 10) The association of operands with operators is called _____.
- (a) assignment
 - (b) precedence
 - (c) binding
 - (d) lexicographic ordering

Answer: C

11) The looping mechanism that always executes at least once is the _____ statement.

- (a) if...else
- (b) do...while
- (c) while
- (d) for

Answer: B

12) A mixture of programming language and human language is known as:

- (a) Algorithms
- (b) Recipes
- (c) Directions
- (d) Pseudocode

Answer: D

13) When the number of repetitions are known in advance, you should use a _____ statement.

- (a) while
- (b) do...while
- (c) for
- (d) None of the above

Answer: C

14) A _____ statement terminates the current iteration of a loop.

- (a) Break
- (b) Continue
- (c) Switch
- (d) Assert

Answer: B

15) Common loop errors are:

- (a) Off-by-one
- (b) Infinite loops
- (c) Both a and b
- (d) None of the above

Answer: C

16) To terminate a program, use the Java statement:

- (a) System.quit(0);
- (b) System.end(0);
- (c) System.abort(0);
- (d) System.exit(0);

Answer: D

- 17) Good debugging techniques include:
- (a) Inserting output statements in your program.
 - (b) Tracing variables
 - (c) Using an IDE debugger
 - (d) All of the above

Answer: D

■ True/False

- 1) An if-else statement chooses between two alternative statements based on the value of a Boolean expression.
Answer: True
- 2) You may omit the else part of an if-else statement if no alternative action is required.
Answer: True
- 3) In a switch statement, the choice of which branch to execute is determined by an expression given in parentheses after the keyword switch.
Answer: True
- 4) In a switch statement, the default case is always executed.
Answer: False
- 5) Not including the break statements within a switch statement results in a syntax error.
Answer: False
- 6) The equality operator (==) may be used to test if two string objects contain the same value.
Answer: False
- 7) Boolean expressions are used to control branch and loop statements.
Answer: True
- 8) The for statement, do...while statement and while statement are examples of branching mechanisms.
Answer: False
- 9) When choosing a sentinel value, you should choose a value that would never be encountered in the input of the program.
Answer: True
- 10) An algorithm is a step-by-step method of solution.
Answer: True

- 11) The three expressions at the start of a for statement are separated by two commas.
Answer: False
- 12) An empty statement is defined as a loop that runs forever.
Answer: False

■ Short Answer/Essay

- 1) What output will be produced by the following code?

```
public class SelectionStatements
{
    public static void main(String[] args)
    {
        int number = 24;
        if(number % 2 == 0)
            System.out.print("The condition evaluated to true!");
        else
            System.out.print("The condition evaluated to false!");
    }
}
```

Answer: The condition evaluated to true!

- 2) What would be the output of the code in #1 if number was originally initialized to 25?
Answer: The condition evaluated to false!

- 3) Write a multi-way if-else statement that evaluates a persons weight on the following criteria: A weight less than 115 pounds, output: Eat 5 banana splits! A weight between 116 pounds and 130 pounds, output: Eat a banana split! A weight between 131 pounds and 200 pounds, output: Perfect! A weight greater than 200 pounds, output: Plenty of banana splits have been consumed!

Answer:

```
if(weight <= 115)
    System.out.println("Eat 5 banana splits!");
else if(weight <= 130)
    System.out.println("Eat a banana split!");
else if(weight <=200)
    System.out.println("Perfect!");
else
    System.out.println("Plenty of banana splits have been consumed!");
```

- 4) Name and describe three types of branching mechanisms and give an example of each.

Answer: Three types of branching mechanisms include the if-else statement, the multi-way if-else statement, and the switch statement. An if-else statement chooses between two alternative statements based on the value of a Boolean expression. If you only need to perform an action if a certain condition is true, then the else part of an if-else statement can be omitted. A multi-way if-else statement allows several conditions to be evaluated. The first condition that evaluates to true, is the branch that is executed. The switch statement is another multi-selection branching mechanism. A switch evaluates a controlling expression and executes the statements in the matching case label.

If – else example:

```
if ( taxableIncome > 30000)
    taxRate = .135;
else
    taxRate = .075;
```

Multi-way if-else example:

```
If(taxableIncome < 10000)
    taxRate = .0425;
else if(taxableIncome < 30000)
    taxRate = .075;
else
    taxRate = .135;
```

Switch example:

```
char myChar = 'b';
switch(myChar)
{
    case 'a':
        System.out.println("It is an A");
        break;

    case 'b':
        System.out.println("It is a B");
        break;

    case default:
        System.out.println("Default case");
        break;
}
```

- 5) Write an if-else statement to compute the amount of shipping due on an online sale. If the cost of the purchase is less than \$20, the shipping cost is \$5.99. If the cost of the purchase over \$20 and at most \$65, the shipping cost is \$10.99. If the cost of the purchase is over \$65, the shipping cost is \$15.99.

Answer:

```
if(costOfPurchase < 20)
    shippingCost = 5.99;
else if((costOfPurchase > 20)&&(costOfPurchase <= 65))
    shippingCost = 10.99;
else
    shippingCost = 15.99;
```

- 6) Evaluate the Boolean equation: $!((6 < 5) \&\& (4 < 3))$

Answer: True

- 7) What is short-circuit evaluation of Boolean expressions?

Answer: Short-circuit evaluation is a mechanism in Java that allows the early termination of the evaluation of a compound Boolean expression. This occurs when two Boolean expressions are joined by the && operator when the first expression evaluates to false. The evaluation short-circuits because no matter what the value of the second expression is, the expression will evaluate to false. When two Boolean expressions are joined by the || operator, if the first expression evaluates to true, then the evaluation will short circuit because the expression will always evaluate to true no matter what the second expression evaluates to.

- 8) Discuss the rules Java follows when evaluating expressions.

Answer: First, Java binds operands with operators by fully parenthesizes the expression using precedence and associativity rules.

Next, Java evaluates expressions from left to right.

Finally, if an operator is waiting for other operands to be evaluated, then that operator is evaluated as soon as its operands have been evaluated.

- 9) Write Java code that uses a do...while loop that prints even numbers from 2 through 10.

Answer:

```
int evenNumber = 2;

do

{

    System.out.println(evenNumber);

    evenNumber += 2;

}while(evenNumber <= 10);
```

- 10) Write Java code that uses a while loop to print even numbers from 2 through 10.

Answer:

```
int evenNumber = 2;

while(evenNumber <= 10)

{

    System.out.println(evenNumber);
```

```
evenNumber += 2;
```

```
}
```

- 11) What is a sentinel value and how is it used?

Answer: A sentinel value is a value that should never be encountered in the input of the program. It is used to control looping mechanisms when the number of repetitions is unknown. When the sentinel value is encountered, the loop terminates. A sentinel value is commonly used with while and do...while statements.

- 12) Write Java code that uses a for statement to sum the numbers from 1 through 50. Display the total sum to the console.

Answer:

```
int sum = 0;

for(int i=1; i <= 50; i++)
{
    sum += i;
}

System.out.println("The total is: " + sum);
```

- 13) What is the output of the following code segment?

```
public static void main(String[] args)

{

    int x = 5;

    System.out.println("The value of x is:" + x);

    while(x > 0)

    {

        x++;
```



```
boolean quit = false;
```

```
System.out.println("This program finds the smallest number"  
+ " in a series of numbers");
```

```
System.out.println("When you want to exit, type Q");
```

```
while(quit != true)
```

```
{
```

```
    System.out.print("Enter a number: ");
```

```
    userInput = keyboard.next();
```

```
    if(userInput.equals("Q") || userInput.equals("q"))
```

```
    {
```

```
        quit = true;
```

```
    }
```

```
    else
```

```
    {
```

```
        int userNumber = Integer.parseInt(userInput);
```

```
        if(userNumber < smallest)

            smallest = userNumber;

        }

    }

    System.out.println("The smallest number is " + smallest);

    System.exit(0);

}

}
```

- 16) Write a complete Java program that uses a for loop to compute the sum of the even numbers and the sum of the odd numbers between 1 and 25.

```
public class sumEvenOdd

{

    public static void main(String[] args)

    {

        int evenSum = 0;

        int oddSum = 0;

        //loop through the numbers

        for(int i=1; i <= 25; i++)

        {

            if(i % 2 == 0)
```

```
{  
  
    //even number  
  
    evenSum += i;  
  
}  
  
else  
  
    {  
  
        oddSum += i;  
  
    }  
  
}  
  
//Output the results  
  
System.out.println("Even sum = " + evenSum);  
  
System.out.println("Odd sum = " + oddSum);  
  
}  
  
}
```