**Question 1**(1 point)

*Saved*

You use the case structure only when a series of decisions is based on a single expression.

Question 1 options:

|  |  |
| --- | --- |
|  | True |
|  | False |

**Question 2**(1 point)

*Saved*

When you need to satisfy two or more criteria to initiate an event in a program, you must make sure that the second decision is made entirely independently of the first decision.

Question 2 options:

|  |  |
| --- | --- |
|  | True\* |
|  | False |

**Question 3**(1 point)

*Saved*

The \_\_\_\_ clause of the decision is the part that executes only when the tested condition in the decision is false.

Question 3 options:

|  |  |
| --- | --- |
|  | If\* |
|  | then |
|  | else |
|  | endif |

**Question 4**(1 point)

*Saved*

The \_\_\_\_ clause is the part of the decision that holds the action or actions that execute when the tested condition in the decision is true.

Question 4 options:

|  |  |
| --- | --- |
|  | else |
|  | endif |
|  | if-then |
|  | Boolean |

**Question 5**(1 point)

*Saved*

A series of nested if statements is also called a \_\_\_\_ if statement.

Question 5 options:

|  |  |
| --- | --- |
|  | Stacked\* |
|  | cascading |
|  | Indented\* |
|  | Waterfall\* |

**Question 6**(1 point)

*Saved*

Most languages allow a specialized selection structure called the \_\_\_\_ structure when there are several distinct possible values for a single variable, and each value requires a different subsequent action.

Question 6 options:

|  |  |
| --- | --- |
|  | cascading |
|  | case |
|  | nested if\* |
|  | short-circuit |

**Question 7**(1 point)

*Saved*

For maximum efficiency, a good rule of thumb in an AND decision is to \_\_\_\_.

Question 7 options:

|  |  |
| --- | --- |
|  | first ask the question that is more likely to be true\* |
|  | first ask the question that is more likely to be false |
|  | rewrite it as an OR decision and ask the question more likely to be true |
|  | rewrite it as an OR decision and ask the question more likely to be false\* |

**Question 8**(1 point)

*Saved*

To avoid confusion, you can use \_\_\_\_statements instead of using AND and OR operators.

Question 8 options:

|  |  |
| --- | --- |
|  | trivial |
|  | nested if |
|  | NOT\* |
|  | range check\* |

**Question 9**(1 point)

*Saved*

For maximum efficiency, a good rule of thumb in an OR decision is to \_\_\_\_.

Question 9 options:

|  |  |
| --- | --- |
|  | first ask the question that is more likely to be true |
|  | first ask the question that is more likely to be false |
|  | rewrite it as an AND decision and ask the question more likely to be true\* |
|  | rewrite it as an AND decision and ask the question more likely to be false |

**Question 10**(1 point)

*Saved*

You can perform a \_\_\_\_ by making comparisons using either the lowest or highest value in a range of values.

Question 10 options:

|  |  |
| --- | --- |
|  | range check |
|  | nested if |
|  | logic check |
|  | trivial expression |

**Question 11**(1 point)

*Saved*

Which of the following is the correct syntax for an if statement?

Question 11 options:

|  |  |
| --- | --- |
|  | if (x < 10) { size = "Small"; } else (x < 20) { size = "Medium"; } |
|  | if (x < 10); { size = "Small"; } else (x < 20) { size = "Medium"; } |
|  | if (x < 10) { size = "Small"; } else { size = "Medium"; } |
|  | if { size = "Small";} else (x < 20) { size = "Medium"; } |

**Question 12**(1 point)

*Saved*

Assuming that the user provides 303 as input, what is the output of the following code snippet?  
int x;  
int y;  
Scanner in = new Scanner(System.in);  
System.out.print("Please enter a number: ");  
y = in.nextInt();  
if (y > 300) {  
    x = y;  
}  
else {  
    x = 0;  
}  
System.out.println("x: " + x);

Question 12 options:

|  |  |
| --- | --- |
|  | x: 0 |
|  | x: 300 |
|  | x: 303 |
|  | There is no output due to compilation errors. |

**Question 13**(1 point)

*Saved*

Assuming that the user provides 99 as input, what is the output of the following code snippet?  
int a;  
int b;  
Scanner in = new Scanner(System.in);  
System.out.print("Please enter a number: ");  
b = in.nextInt();  
if (b > 300) {  
    a = b;  
}  
else {  
    a = 0;  
}  
System.out.println("a: " + a);

Question 13 options:

|  |  |
| --- | --- |
|  | a: 0 |
|  | a: 99 |
|  | a: 100 |
|  | a: 300 |

**Question 14**(1 point)

*Saved*

The following code snippet contains an error. What is the error?  
int cost = 100;  
if (cost > 100);{  
    cost = cost – 10;  
}  
System.out.println("Discount cost: " + cost);

Question 14 options:

|  |  |
| --- | --- |
|  | Syntax error (won't compile)\* |
|  | Logical error: use of an uninitialized variable\* |
|  | Logical error: if statement has do-nothing statement after if condition |
|  | Logical error: assignment statement does not show equality |

**Question 15**(1 point)

*Saved*

What is the output of the following code snippet if the input is 25?  
int i;  
Scanner in = new Scanner(System.in);  
System.out.print("Enter a number: ");  
i = in.nextInt();  
if (i > 25) {  
    i++;  
}  
else {  
    i--;  
}  
System.out.print(i);

Question 15 options:

|  |  |
| --- | --- |
|  | 24 |
|  | 25 |
|  | 26 |
|  | 27 |

**Question 16**(1 point)

*Saved*

What is the syntax error in the following if statement?  
double count = 15.0;  
if (count / 3.0) {  
    System.out.println("The value of count is ");  
}

Question 16 options:

|  |  |
| --- | --- |
|  | There should be an "else" condition |
|  | The condition does not evaluate to a Boolean value |
|  | The variable count should be part of the string |
|  | It is never possible to use the "/" operator in an if statement |

**Question 17**(1 point)

*Saved*

Consider the following code snippet. What is the potential problem with the if statement?  
double average;  
average = (g1 + g2 + g3 + g4) / 4.0;  
if (average == 90.0){  
     System.out.println("You earned an A in the class!");  
}

Question 17 options:

|  |  |
| --- | --- |
|  | Using == to test the double variable average for equality is error-prone. |
|  | The conditional will not evaluate to a Boolean value. |
|  | The assignment operator should not be used within an if-statement conditional.\* |
|  | Literals should never be used in if statement conditionals.\* |

**Question 18**(1 point)

*Saved*

What is the output of the following code snippet?  
final int EXPERIENCE = 5;  
int age = 25;  
if ((age + EXPERIENCE) > 30) {  
    System.out.println("You are wise!");  
}  
else {  
    System.out.println("You have much to learn!");  
}

Question 18 options:

|  |  |
| --- | --- |
|  | There is no output due to compilation errors. |
|  | You are wise! |
|  | You have much to learn! |
|  | You are wise! You have much to learn! |

**Question 19**(1 point)

*Saved*

If the user entered the value 5, what is the output from the following code snippet?  
Scanner input = new Scanner(System.in);  
int option = input.nextInt();  
switch(option){  
case 5:  
    option = option + 1;  
case 2:  
    option = option + 4;  
case 3:  
    option = option + 3;  
default:  
    option = option + 2;  
}  
System.out.println(option);

Question 19 options:

|  |  |
| --- | --- |
|  | 4 |
|  | 5 |
|  | 8 |
|  | 15 |

**Question 20**(1 point)

*Saved*

Which of the following performs the same way as the switch statement below?  
switch (value) {  
case 1:  
    System.out.print ("Small");  
    break;  
case 10:  
    System.out.print ("Large");  
    break;  
default:  
    System.out.print ("Other");  
    break;  
}

Question 20 options:

|  |  |
| --- | --- |
|  | if (value == 1) System.out.print("Small"); else if (value == 10) System.out.print("Large"); else System.out.print("Other"); |
|  | if (value == 1) System.out.print("Small"); if (value == 10) System.out.print("Large"); else System.out.print("Other"); |
|  | if (value >= 10) System.out.print("Large"); else System.out.print("Other"); |
|  | if (value == 10) System.out.print("Large"); else System.out.print("Other"); |

**Question 21**(1 point)

*Saved*

What is the Method Call Stack?

Question 21 options:

|  |  |
| --- | --- |
|  | An allocation of memory, used to store methods and their data. |
|  | An allocation of memory, used to store reference types (objects).\* |
|  | Where methods go when the compiler executes them. |

**Question 22**(1 point)

*Saved*

What is the Heap?

Question 22 options:

|  |  |
| --- | --- |
|  | An allocation of memory, used to store methods and their data.\* |
|  | An allocation of memory, used to store reference types (objects). |
|  | Where objects go when they are no longer needed. |

**Question 23**(1 point)

*Saved*

What is a method call, within the Method Call Stack, referred to as?

Question 23 options:

|  |  |
| --- | --- |
|  | a plate |
|  | a counter |
|  | stack frame |

**Question 24**(1 point)

*Saved*

What typically has more memory available, Method Call Stack or Heap memory?

Question 24 options:

|  |  |
| --- | --- |
|  | Method Call Stack |
|  | Heap memory |
|  | Both have the same amount. |

**Question 25**(1 point)

*Saved*

Where are reference types (objects) stored in memory?

Question 25 options:

|  |  |
| --- | --- |
|  | Method Call Stack |
|  | Heap |
|  | Garbage Collector stores them |