

**Started on** Wednesday, 25 September 2024, 8:49 PM

**State** Finished

**Completed on** Wednesday, 25 September 2024, 9:06 PM

**Time taken** 16 mins 58 secs

**Marks** 7.00/10.00

**Grade** 70.00 out of 100.00

## Question

### 1

Complete

Mark 1.00 out of 1.00

Which of the following statements best distinguishes between supervised learning and unsupervised learning?

Select one or more:

- ☐ a. In supervised learning, the algorithm groups data into clusters, while in unsupervised learning, the algorithm makes predictions based on labeled data.
- ☐ b. Unsupervised learning can only work with numerical data, while supervised learning can handle both numerical and categorical data.
- ☒ c. Supervised learning requires labeled data, whereas unsupervised learning does not.
- ☐ d. Supervised learning is typically used for anomaly detection, whereas unsupervised learning is used for classification tasks.

## Question

### 2

Complete

Mark 0.00 out of 1.00

What will be the output of the above code when the infix expression " $a+b*(c^d-e)^{(f+g*h)-i}$ " is converted to postfix?

Select one or more:

- ☐ a.  $abcd^e-fgh+^{*+}i-$
- ☒ b.  $ab+cd^e-fgh^{*+}i-$
- ☐ c.  $abcd^e-fgh^{*+}^{*+}i-$
- ☐ d.  $abc^de-fgh^{*+}^{*+}i-$

## Question

# 3

Complete

Mark 1.00 out of 1.00

Choose the correct statement about the following code:

```
1: interface HasExoskeleton {  
2:     abstract int getNumberOfSections();  
3: }  
4: abstract class Insect implements HasExoskeleton {  
5:     abstract int getNumberOfLegs();  
6: }  
7: public class Beetle extends Insect {  
8:     int getNumberOfLegs() { return 6; }  
9: }
```

Select one or more:

- ☐ a. The code will not compile because of line 4.
- ☐ b. The code will not compile because of line 2.
- ☒ c. The code will not compile because of line 7.
- ☐ d. It compiles but throws an exception at runtime.
- ☐ e. It compiles and runs without issue.

## Question

# 4

Complete

Mark 0.00 out of 1.00

Consider the following code that calculates the factorial of a number using recursion:

```
public class Factorial {  
    public static int factorial(int n) {  
        if (n == 0 || n == 1) {  
            return 1;  
        } else {  
            return n * factorial(n - 1);  
        }  
    }  
}  
  
public static void main(String[] args) {  
    System.out.println(factorial(5));  
}
```

If the recursive call depth equals the stack depth, how many stack frames will be used to calculate factorial(5)?

Select one or more:

- ☒ a. 5
- ☐ b. 10
- ☐ c. 6
- ☐ d. 4

## Question

# 5

Complete

Mark 1.00 out of 1.00

Which of the following compile? (Choose all that apply)

Select one or more:

- ☐ a. **public void moreE(String[] values, ...int nums) {}**
- ☒ b. **public void moreA(int... nums) {}**
- ☒ c. **public void moreG(String[] values, int[] nums) {}**
- ☐ d. **public void moreF(String... values, int[] nums) {}**
- ☐ e. **public void moreC(int... nums, String values) {}**
- ☐ f. **public void moreD(String... values, int... nums) {}**
- ☒ g. **public void moreB(String values, int... nums) {}**

## Question

# 6

Complete

Mark 1.00 out of 1.00

Consider the following Java code:

java

Copy code

```
import java.util.Stack;

public class Main {
    public static void main(String[] args) {
        Stack<Integer> stack = new Stack<>();
        stack.push(1);
        stack.push(2);
        stack.push(3);

        for (int i : stack) {
            if (i == 2) {
                stack.pop(); // Line A
            }
        }

        System.out.println(stack);
    }
}
```

What will happen when the program reaches Line A?

Select one or more:

- ☐ a. A `NullPointerException` will be thrown.
- ☒ b. A `ConcurrentModificationException` will be thrown.
- ☐ c. The stack will correctly remove the element 2 and print [1, 3].
- ☐ d. The program will remove the top of the stack (3), and print [1, 2].

## Question

# 7

Complete

Mark 1.00 out of 1.00

What is the result of the following code?

```
7: StringBuilder sb = new StringBuilder();  
8: sb.append("aaa").insert(1, "bb").insert(4, "ccc");  
9: System.out.println(sb);
```

Select one or more:

- ☐ a. bbaaccba
- ☒ b. abbacca
- ☐ c. abbaaccc
- ☐ d. bbaaaccc
- ☐ e. An exception is thrown.
- ☐ f. The code does not compile

## Question

# 8

Complete

Mark 1.00 out of 1.00

Given the following class, which of the following calls print out Blue Jay? (Choose all that apply)

```
public class BirdDisplay {  
    public static void main(String[] name) {  
        System.out.println(name[1]);  
    }  
}
```

Select one or more:

- ☒ a. java BirdDisplay Sparrow "Blue Jay"
- ☐ b. java BirdDisplay Blue Jay Sparrow
- ☐ c. java BirdDisplay.class "Blue Jay" Sparrow
- ☐ d. Does not compile
- ☐ e. java BirdDisplay "Blue Jay" Sparrow
- ☐ f. java BirdDisplay.class Sparrow "Blue Jay"
- ☐ g. java BirdDisplay Sparrow Blue Jay

## Question

# 9

Complete

Mark 0.00 out of  
1.00

What change would allow the following code snippet to compile? (Choose all that apply)

```
3: long x = 10;
```

```
4: int y = 2 * x;
```

Select one or more:

- ☒ a. Change the data type of y on line 4 to long.
- ☐ b. Change the data type of x on line 3 to short.
- ☐ c. No change; it compiles as is
- ☒ d. Cast 2 \* x on line 4 to int.
- ☐ e. Change the data type of y on line 4 to short.
- ☒ f. Cast x on line 4 to int.

## Question

# 10

Complete

Mark 1.00 out of 1.00

Consider the following Java code using a stack in a multi-threaded environment:

```
import java.util.Stack;

class Worker extends Thread {
    private Stack<Integer> stack;

    Worker(Stack<Integer> stack) {
        this.stack = stack;
    }

    public void run() {
        for (int i = 0; i < 5; i++) {
            stack.push(i);
            System.out.println(Thread.currentThread().getName() + " pushed: " + i);
        }
    }
}

public class Main {
    public static void main(String[] args) throws InterruptedException {
        Stack<Integer> stack = new Stack<>();

        Worker worker1 = new Worker(stack);
        Worker worker2 = new Worker(stack);

        worker1.start();
        worker2.start();

        worker1.join();
        worker2.join();

        System.out.println("Final stack: " + stack);
    }
}
```

What is a possible output of the above program?

Select one or more:

- ☒ a. The final stack will contain the numbers 0 through 4 added by both threads, in no particular order.
- ☐ b. A `ConcurrentModificationException` will be thrown.
- ☐ c. The program will throw a `NullPointerException`.
- ☐ d. The program will deadlock.