Started on	Thursday, 2 January 2025, 9:50 AM
State	Finished
Completed on	Thursday, 2 January 2025, 11:33 AM
Time taken	1 hour 43 mins
Grade	Not yet graded

Question **1**

Correct

Mark 2.00 out of 2.00

Which of the following methods is used to create a new thread in Java?

a. run()

b. start()

c. execute()

d. begin()

Your answer is correct.

The correct answer is:
start()

Question **2**

Correct

Mark 2.00 out of 2.00

The finally block is optional in Java exception handling.

Select one:

True ✔

False

The correct answer is 'True'.

Question **3**

Correct

Mark 2.00 out of 2.00

In Java, `String` objects are immutable.

Select one:

- True ✔
- False

The correct answer is 'True'.

Question **4**

Complete

Marked out of 6.00

Discuss the difference between checked and unchecked exceptions in Java. Provide examples of each and expla

Checked exceptions are exceptions that the compiler forces you to handle. You must either catch them using
declare them in the method using the `throws` keyword.
Unchecked exceptions are exceptions that are not required to be handled - like errors that are the result of m
pointer dereferencing or array index out-of-bounds.

Question **5**

Complete

Marked out of 6.00

Describe the lifecycle of a Java program from source code to execution. Include the roles of the Java Compiler a (JVM).

Java LifeCycle:
1. Source Code: we write Java source code in `.java` files.
2. Compilation (javac): The `javac` compiler converts the `.java` file into bytecode (`.class` files)
3. Bytecode Verification: The bytecode is verified by the JVM (a componenet called class loader/verifier) to ens violations of the security rules or other errors.
4. Execution: The JVM executes the bytecode. It chooses either interprets the bytecode or compiles it into nati **JIT (Just-In-Time)** compiler for better performance OR to compile it Ahead of Time (AOT)

Question **6**

Correct

Mark 2.00 out of 2.00

In Java, threads can be created by extending the `Thread` class or implementing the `Runnable` interface.

Select one:

- (•) True ✔
- ( ) False

The correct answer is 'True'.

Question **7**

Complete

Marked out of 6.00

Implement a Java class `Circle` with a private field `radius` (double). Include methods to calculate the area and ci
circle.

```
public class Circle {

   private double radius;

   public Circle(double radius) {
      if (radius < 0) {
         throw new IllegalArgumentException("Radius must be positive");
      }
      this.radius = radius;
   }

//IF NEEDED - GETERS & SETTERS GO HERE

   public double calculateArea() {
      return Math.PI * radius * radius;
```

Question **8**

Correct

Mark 2.00 out of 2.00

What is the output of the following code?

System.out.println(10 + 20 + "Java" + 30 + 40);

- a. 30Java3040
- b. 30Java70
- c. 30Java30 40
- d. 1020Java3040

Your answer is correct.

The correct answer is:
30Java3040

Question **9**

Complete

Marked out of 6.00

Create a Java program that reads a text file named `input.txt`, counts the number of words in the file, and print

String filePath = "input.txt";

BufferedReader br = new BufferedReader(new FileReader(filePath))

```
package src;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class HelloWorld {
  public static void main(String[] args) {
    countWordsFile();
  }

  public static void countWordsFile() {
    // MAKE SURE THE WORDS IN THE FILE ARE SEPERATED BY ONE SPACE ONLY, OTHERWISE EACH EXTRA SPA
WORD TOO
    try (BufferedReader reader = new BufferedReader(new FileReader("input.txt"))) {
```

Question **10**

Correct

Mark 2.00 out of 2.00

What is the purpose of the `super` keyword in Java?

- a. To create a new thread
- b. To finalize an object
- c. To refer to the current object
- d. To call the superclass constructor

Your answer is correct.

The correct answer is:
To call the superclass constructor

Question **11**

Complete

Marked out of 6.00

Explain the concept of Object-Oriented Programming (OOP) in Java. How do the four main principles of OOP—e inheritance, polymorphism, and abstraction—manifest in Java?

Object-Oriented Programming is a programming paradigm based on the concept of objects, which are instan allows you to model real-world entities, so thw SW we write will behave like objects in the real world. it is base
1. Encapsulation - restricts direct access to some of the object's components which improves security and red interference with its data.
2. Inheritance - allows a class to inherit properties and behaviors from another class- makes the system more
3. Polymorphism - allows one entity (method or object) to take on multiple forms, so behavior is customized p
4. Abstraction- a concept of hiding the complex implementation details , focusing on high-level operations, an implement some attributes.

Question **12**

Complete

Marked out of 6.00

Write a Java method `reverseArray(int[] array)` that takes an array of integers and returns a new array with tl order.

```
package src;

public class Arr {

  public static int[] revArr(int[] arr) {

     int[] rev = new int[arr.length];
     for (int i = 0; i < arr.length; i++) {
        rev[i] = arr[arr.length - 1 - i];
     }
     return rev;
  }

  public static void main(String[] args) {
     int[] arr = {1, 2, 3, 4};
```

Question **13**

Correct

Mark 2.00 out of 2.00

Java supports multiple inheritance with classes

Select one:

- True
- False ✔

The correct answer is 'False'.

Question **14**

Correct

Mark 2.00 out of 2.00

An abstract class can be instantiated in Java.

Select one:

- True
- False ✔

The correct answer is 'False'.

Question **15**

Correct

Mark 2.00 out of 2.00

Which of the following is not a primitive data type in Java?

- a. int
- b. double
- c. boolean
- d. String

Your answer is correct.

The correct answer is:
String

Question **16**

Complete

Marked out of 6.00

Explain the use of the `static` keyword in Java. Provide examples of its application in variables, methods, and bl

a static keyword declared with an entity (variables, methods, and blocks.) makes that entity belong to the cla
it symbolizes that it will be considered as an attribute of the class, and not of any of the class's instances, mea
to access it statically in the ways listed below.
also important - a static field will get a default initialization by JVM (implicitly, for example a static int will be 0

for variables:
private static int counterID; --common for counting number of instances created from this class

for metoods:
public static void sayHello(); --some static method that will be accessed by myClass.sayHello();

for a static block:
static {
// one time runner
}

Question **17**

Complete

Marked out of 6.00

Write a Java program that creates a thread which prints "Hello from Thread" five times, pausing for one second

```
package src;
public class Hello {

  public static void main(String[] args) {
     Thread thread = new Thread(() -> {
        for (int i = 0; i < 5; i++) {
           System.out.println("Hello from Thread");
           try {
              Thread.sleep(1000);
           } catch (InterruptedException e) {
              System.err.println("Thread Error");
           }
        }
     });
```

Question **18**

Correct

Mark 2.00 out of 2.00

What is the default value of a boolean variable in Java?

- a. 0
- b. false
- c. true
- d. null

Your answer is correct.

The correct answer is:
false

Question **19**

Complete

Marked out of 6.00

Describe how Java handles memory management, particularly focusing on the heap and stack.

Memory management in Java is divided into heap & stack.
1. Heap: used for dynamic memory allocation, primarily storing un-primitive data like objects and arrays.
When you create a new object using new, it is stored in the heap.
The heap is garbage collected, meaning the Java Garbage Collector (GC) automatically clears it if it detects
referenced.
2. Stack: used for method execution and local variables.
Each time a method is called, a new stack frame is created containing the method's local variables and re
Stack memory is automatically managed: it is allocated when a method is invoked and deallocated when

Question **20**

Complete

Marked out of 6.00

Explain the significance of the `main` method in a Java application. Can a Java program run without it? Why or wh

he main method serves as the entry point for the application. When a Java program is executed, the JVM look
in the specified class that we are running in the command line (so there must not be 2 main signatures in a si

public static void main(String[] args):
public: It must be public so that the JVM can access it from outside the class.
static: It is static, meaning it can be called without creating an instance of the class.
void: It does not return any value.
String[] args: This allows the program to accept command-line arguments when it is run.

Question **21**

Complete

Marked out of 6.00

Create a Java method `isPrime(int number)` that returns `true` if the given number is a prime number, and `false`

```
package src;

public class Prime {
    public static void main(String[] args) {

        int[] testArr = {5, 10, 15, 56, 17};
        for (int i = 0; i < testArr.length; i++) {
            if (isPrime(testArr[i])) {
                System.out.println(testArr[i] + " YES");
            } else {
                System.out.println(testArr[i] + " NOT PRIME");
            }
        }
    }
```

Question **22**

Correct

Mark 2.00 out of 2.00

The **final** keyword in Java means that the variable or method belongs to the instance of the class.

Select one:

- True
- False ✔

The correct answer is 'False'.

Question **23**

Complete

Marked out of 6.00

Implement a Java class `Car` with private fields `make`, `model`, and `year`. Include methods to start the car, stop the c details.

```
package src;

// Implement a Java class Car with private fields make, model, and year. Include methods to start the car, stop
its details.

public class Car {

    private String make;
    private String model;
    private int year;

    public Car(String make, String model, int year) {
        this.make = make;
        this.model = model;
        this.year = year;
```

Question **24**

Correct

Mark 2.00 out of 2.00

Which of the following is not a valid access modifier in Java?

- a. public
- b. private
- c. static
- d. protected

Your answer is correct.

The correct answer is:
static

Question **25**

Incorrect

Mark 0.00 out of 2.00

`ArrayList` in Java can store primitive data types like `int` and `char`.

Select one:

- True
- False ✖

The correct answer is 'True'.

Question **26**

Correct

Mark 2.00 out of 2.00

How can you make a class in Java immutable?

- a. By making all fields private and final
- b. All of the above
- c. By declaring the class as final
- d. By not providing setter methods

Your answer is correct.

The correct answer is:
All of the above

◄ Demo test.

Jump to...