# FULL STACK DEVELOPMENT – WORKSHEET 2

Q1. Java method overloading implements the OOPS concept

A. Encapsulation

B. Inheritance

**C. Polymorphism**

D. Abstraction

Q2. Data members and member functions of a class are private by default.

**A. True**

B. False

C. Depend on code

D. None

Q3. Which of the following functions can be inherited from the base class?

A. Constructor

**B. Static**

C. All

D. None

Q4. Identify the feature, which is used to reduce the use of nested classes.

A. Binding

**B. Abstraction**

C. Inheritance

D. None

Q5. Which concept of Java is achieved by combining methods and attributes into a class?

**A. Encapsulation**

B. Inheritance

C. Polymorphism

D. Abstraction

Q6. Which of the following declarations does not compile?

**A. double num1, int num2 = 0;**

B. int num1, num2;

C. int num1, num2 = 0;

D. int num1 = 0, num2 = 0;

Q7. Which of these interfaces must contain a unique element?

**A. Set**

B. List

C. Array

D. Collection

8. The given Java code defines two classes, **T and Main** Class **T** has an instance variable **t** with an initial value of 20.

The given Java code defines two classes, **T** and **Main**.

Class **T** has an instance variable **t** with an initial value of 20.

Now let's analyze the code:

When you create an instance of class `T` using `T t1 = new T();`, it initializes the `t` variable to 20 because that's its initial value.

So, the output of the code will be:

A. 20

Explanation: The code creates an instance of class `T`, and since the `t` variable in class `T` is initialized to 20, when you print `t1.t`, it will output 20.

9. The output of the given Java program will be:

A. BINGO

Explanation:

1. The program defines a class named **Hello** with the **main** method.
2. In the **main** method, it uses **System.out.println("BINGO");** to print the string "BINGO" to the console.
3. Java is case-sensitive, so it will print "BINGO" exactly as it is written in the **System.out.println** statement.

Therefore, the output will be "BINGO" with uppercase letters. Option A is correct.

10. The output of the given Java program will be:

A. Compilation Error

Explanation:

1. The program defines a class named `variable_scope` with the `main` method.

2. Inside the `main` method, there is an integer variable `x` declared and initialized with the value 5.

3. Then, there is a block of code enclosed in curly braces where another integer variable `y` is declared and initialized with the value 6.

4. Inside this block, it prints the values of `x` and `y` using `System.out.print(x + " " + y);`. This will print "5 6" to the console.

5. Outside the block, it attempts to print the values of `x` and `y` again using `System.out.println(x + " " + y);`. However, this will result in a compilation error because the variable `y` is not in scope outside of the block where it was declared.

Therefore, the correct answer is A. Compilation Error.

11. The output of the given Java code will be:

A. abc

Explanation:

1. The program defines a class named `String_demo` with the `main` method.
2. Inside the `main` method, it creates an array of characters `chars` containing the characters 'a', 'b', and 'c'.
3. It then creates a new `String` object `s` by passing the `chars` array to its constructor. This constructor converts the character array into a string.
4. Finally, it prints the value of the `s` string using `System.out.println(s);`, which will output "abc" because the `chars` array is converted into a string, and the string "abc" is stored in the variable `s`.

12. Therefore, the correct answer is A. abc.

---

1. Class `A` is declared as `final`, which means it cannot be extended. Therefore, the `class B extends A` would result in a compilation error. To fix this, we remove the `final` keyword from class `A`.
2. The `System.out.println(j + " " + i);` statement is placed directly inside the `class B`, which is not allowed. Instead, it should be placed inside a method. In this case, I added a method called `display()` inside class `B` to print the values of `j` and `i`.

With these corrections, the code will compile successfully.

Now, the output of the corrected code will depend on the values of `j` and `i` when an object of class `B` is created and the `display()` method is called. Since there are no initializations provided in the code, the output will depend on the default values for `int` variables, which is `0`.

So, the output will be:

A. 0 0

13. The given Java program will result in a compilation error because it has two methods with the same name `getData` and the same parameter types. Method overloading in Java is based on the method name and the parameter types. Since these two methods have the same name and parameter types (no parameters), they cannot be distinguished by the compiler, leading to a compilation error.

So, the correct answer is:

D. Compilation Error

14.  The output of the given Java program will be:

C. [5 2]

Here's the breakdown of how we get this output:

1. We create a new `Test` object with `new Test(10)`. This calls the constructor `public Test(int x)` with `x` set to 10. Inside the constructor, `x` is set to 4, but `end` is set to `x`, which is 10.

2. After creating the object, we immediately call the `fly` method on it with `fly(5)`.

3. Inside the `fly` method, it prints `end-start` and `distance`.

   - `end-start` is `10 - 2`, which is 8.

   - `distance` is 5.

So, the output is "8 5" as given in option C.

15. The output of the given Java program will be:

C. false true

Here's the explanation:

1. `String john = "john";` creates a string literal "john" and assigns it to the variable `john`.

2. `String jon = new String(john);` creates a new string object `jon` using the `new` keyword and the value of `john`. This will result in a new string object even though the content is the same as the string literal "john".

3. `(john==jon)` compares the references of `john` and `jon`. Since `john` is a string literal, it is stored in the string pool, and `jon` is a new string object, they will have different references. Therefore, `(john==jon)` is `false`.

4. `(john.equals(jon))` compares the contents of `john` and `jon`. Both strings have the same content ("john"), so `(john.equals(jon))` is `true`.

So, the output is "false true" as given in option C.

16. The following code creates reference variables and objects as follows:

1. `Student studentName, studentId;` declares two reference variables of type `Student` but does not create any objects. They are just references.

2. `studentName = new Student();` creates a new object of the `Student` class and assigns its reference to the `studentName` reference variable. So, one object is created here.

3. `Student stud_class = new Student();` creates another new object of the `Student` class and assigns its reference to the `stud_class` reference variable. So, one more object is created here.

So, in total, there are:

B. Two reference variables and two objects are created.

17. Write a java program to check even or odd number.

```
import java.util.Scanner;
public class EvenOddChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        if (isEven(number)) {
            System.out.println(number + " is an even number.");
        } else {
            System.out.println(number + " is an odd number.");
        }
        scanner.close();
    }
    public static boolean isEven(int number) {
        return number % 2 == 0;
    }
}
```

18. Write a java program to find average of two numbers:

```
import java.util.Scanner;

public class AverageCalculator {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the first number: ");

        double number1 = scanner.nextDouble();

        System.out.print("Enter the second number: ");

        double number2 = scanner.nextDouble();

        double average = calculateAverage(number1, number2);

        System.out.println("The average of " + number1 + " and " + number2 + "
is: " + average);

       scanner.close();

    }

    public static double calculateAverage(double num1, double num2) {

        return (num1 + num2) / 2.0;

    }

}
```

19. Write a java program to swap two numbers

```
import java.util.Scanner;
public class SwapNumbers {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        int num1 = scanner.nextInt();
        System.out.print("Enter the second number: ");
        int num2 = scanner.nextInt();
        System.out.println("Before swapping:");
        System.out.println("First number: " + num1);
        System.out.println("Second number: " + num2);
        swapNumbers(num1, num2);
        System.out.println("After swapping:");
        System.out.println("First number: " + num1);
        System.out.println("Second number: " + num2);
        scanner.close();
    }
    public static void swapNumbers(int a, int b) {
        int temp = a;
```

```
a = b;
        b = temp;
    }
}
```

20. Write a number whether a number is prime is not.

```
import java.util.Scanner;


public class PrimeNumberChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        if (isPrime(number)) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }
        scanner.close();
    }
    public static boolean isPrime(int number) {
        if (number <= 1) {
            return false;
        }
        // Check for factors from 2 to the square root of the number
```

```
        for (int i = 2; i * i <= number; i++) {
            if (number % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

21. Write a java program to find table of n

```
import java.util.Scanner;
public class MultiplicationTable {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number to generate its multiplication table: ");
        int n = scanner.nextInt();
        System.out.print("Enter the limit for the multiplication table: ");
        int limit = scanner.nextInt();
        printMultiplicationTable(n, limit);
        scanner.close();
    }
    public static void printMultiplicationTable(int n, int limit) {
        System.out.println("Multiplication table of " + n + " up to " + limit + "
is:");
        for (int i = 1; i <= limit; i++) {
            int result = n * i;
            System.out.println(n + " x " + i + " = " + result);
        }
    }}
```

21. Write a java program to find the largest three numbers

```
import java.util.Scanner;
public class LargestOfThree {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        double num1 = scanner.nextDouble();
        System.out.print("Enter the second number: ");
        double num2 = scanner.nextDouble();
        System.out.print("Enter the third number: ");
        double num3 = scanner.nextDouble();
        double largest = findLargest(num1, num2, num3);
        System.out.println("The largest number is: " + largest);
        scanner.close();
    }
    public static double findLargest(double a, double b, double c) {
        if (a >= b && a >= c) {
            return a;
        } else if (b >= a && b >= c) {
            return b;
```

```
        } else {
            return c;
        }
```

23. Write a java program to calculate Simple Interest

Sure, here's a Java program that calculates simple interest based on user input:

```java
import java.util.Scanner;

public class SimpleInterestCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the principal amount: ");
        double principal = scanner.nextDouble();

        System.out.print("Enter the annual interest rate (as a percentage): ");
        double rate = scanner.nextDouble();

        System.out.print("Enter the time period (in years): ");
```

```
        double time = scanner.nextDouble();

        double simpleInterest = calculateSimpleInterest(principal, rate, time);

        System.out.println("Simple Interest: " + simpleInterest);

        scanner.close();
    }

    public static double calculateSimpleInterest(double principal, double rate,
double time) {
        // Convert rate from percentage to decimal
        rate /= 100.0;

        // Calculate simple interest
        double simpleInterest = principal * rate * time;

        return simpleInterest;
    }
}
```

```
    }
}
```

24. Write a java program to calculate Area and perimeter of Rectangle

```
import java.util.Scanner;

public class RectangleAreaPerimeterCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the length of the rectangle: ");
        double length = scanner.nextDouble();
        System.out.print("Enter the width of the rectangle: ");
        double width = scanner.nextDouble();
        double area = calculateRectangleArea(length, width);
        double perimeter = calculateRectanglePerimeter(length, width);

        System.out.println("Area of the rectangle: " + area);
        System.out.println("Perimeter of the rectangle: " + perimeter);
        scanner.close();
    }
```

```
    public static double calculateRectangleArea(double length, double width)
{

        return length * width;

    }

    public static double calculateRectanglePerimeter(double length, double
width) {

        return 2 * (length + width);

    }

}
```

25. Write a java program to check whether character is vowel or consonant

```
import java.util.Scanner
public class VowelConsonantChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter a character: ");
        char character = scanner.next().charAt(0);


        if (isVowel(character)) {
            System.out.println(character + " is a vowel.");
        } else {
            System.out.println(character + " is a consonant.");
        }
        scanner.close();
    }
    public static boolean isVowel(char c) {
        // Convert the character to lowercase for case-insensitive comparison
```

```
        char lowercaseChar = Character.toLowerCase(c);

        // Check if the character is one of the vowels

        return lowercaseChar == 'a' || lowercaseChar == 'e' || lowercaseChar
== 'i' || lowercaseChar == 'o' || lowercaseChar == 'u';

    }

}
```

---

---