FULL STACK DEVELOPMENT – WORKSHEET 3

Q1. Which one of the following is not a Java feature?

Ans: (B) Use of pointers

Q2. Which of these cannot be used for a variable name in Java?

Ans: (C) keywords

Q3. Which of the following is a superclass of every class in Java?

Ans: (C) Object Class

Q4. Which one is a valid declaration of a boolean?

Ans: (C) boolean b3 = false

Q5. Which is the modifier when there is none mentioned explicitly?

Ans: (D) default

Q6. All the variables of interface should be?

Ans: (C) public, static and final

Q7. Which of these data types is used to store command line arguments?

Ans: (C) String

Q8. How many arguments can be passed to main()?

Ans: (D) None of the mentioned

Q9. What will be the output of the following Java program, Command line execution is done as – "java Output This is a command Line"?

```
class Output {
public static void main (String args[]) {
System.out.print(args[0]);
   }
}
```

Ans: (C) This

Q10.What is the value of "d" in the following Java code snippet? double d = Math.round (2.5 + Math.random());

Ans: (B) 3

Q11.Which of these methods is a rounding function of Math class?

Ans: (D) all of the mentioned

Q12. Standard output variable 'out' is defined in which class?

Ans: (D) System

Q13.What will be the output of the following Java program?

```
class main_class {
public static void main(String args[]) {
int x = 9; if (x == 9) {
int x = 8; System.out.println(x);
   }
 }
}
```

Ans: (C) Compilation error

Q14. Which of these is the method which is executed first before execution of any other thing takes place in a program?

Ans: (B) static method

Q15. Which of these can be used to differentiate two or more methods having the same name?

Ans: (D) all of the mentioned

Q16. What will be the output of the following Java program?

```
class Output {
static void main(String args[]) {
int x , y = 1; x = 10; if(x != 10 && x / 0 == 0)
System.out.println(y);
else System.out.println(++y);
 }
}
```

Ans: (B) 2

Q17.What will be the output of the following Java program?

```
class area {
 int width;
int length;
int height; area()
{
width = 5;
length = 6;
height = 1;
}
void volume() {
volume = width * height * length;
 }
}
class cons_method {
public static void main(String args[]) {
area obj = new area();
obj.volume();
System.out.println(obj.volume);
  }
}
```

Ans: (D) 30

Q18. Write Syntax to create/define java methods.

Ans:

```
[modifiers] returnType methodName(parameters) {
   // method body
}
```

Q19. Write a java program following instructions

A. Make a class Addition

a. initialize sum as 0

b. make add Two Int method taking two int parameters a,b. make sum = a+b. Return Sum

Ans:

```
public class Addition {
    // Initialize sum as 0
    private int sum = 0;

    // Method to add two integers and return the sum
    public int addTwoInt(int a, int b) {
        sum = a + b;
        return sum;
    }

    // Main method to test the addTwoInt method
    public static void main(String[] args) {
        Addition addition = new Addition();
        int result = addition.addTwoInt(5, 7);
        System.out.println("The sum of 5 and 7 is: " + result);
    }
}
```

(B) . define class as Method Call. Define main method

a. Create object of class Addition

b. call method using instance of object

c. Print sum

Ans:

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

```
        // Create an object of class Addition
        Addition addition = new Addition();

        // Call the addTwoInt method using the instance of Addition
        int sum = addition.addTwoInt(5, 7);

        // Print the sum
        System.out.println("The sum of 5 and 7 is: " + sum);
    }
}

class Addition {
    // Initialize sum as 0
    private int sum = 0;

    // Method to add two integers and return the sum
    public int addTwoInt(int a, int b) {
        sum = a + b;
        return sum;
    }
}
```

Q20. Write a java program following instructions

A. Define a class Example

a. Define two instance variables number and name

b. Define accessor (getter) methods

c. Define mutator (setter) methods

d. define method printDetails —-> print name and number

Ans:

```
public class Example {
    // Define two instance variables
    private int number;
    private String name;

    // Getter method for number
    public int getNumber() {
        return number;
    }

    // Setter method for number
    public void setNumber(int number) {
        this.number = number;
    }

    // Getter method for name
    public String getName() {
        return name;
    }
```

```
    // Setter method for name
    public void setName(String name) {
        this.name = name;
    }

    // Method to print details
    public void printDetails() {
        System.out.println("Name: " + name);
        System.out.println("Number: " + number);
    }

    public static void main(String[] args) {
        // Create an instance of Example
        Example example = new Example();

        // Set values using setter methods
        example.setName("Alice");
        example.setNumber(42);

        // Print details using printDetails method
        example.printDetails();
    }
}
```

(B) Define public class Demo (Main Class)

a. Define main method

b. Make Instance/object of example class

c. set number and name using instance created as 123 and Your name.

d. call printDetails method using instance

Ans:

```
public class Demo {
    public static void main(String[] args) {
        // Create an instance of Example
        Example example = new Example();

        // Set values using setter methods
        example.setNumber(123);
        example.setName("Your Name");

        // Print details using printDetails method
        example.printDetails();
    }
}

class Example {
    // Define two instance variables
    private int number;
    private String name;

    // Getter method for number
    public int getNumber() {
        return number;
```

```
    }

    // Setter method for number
    public void setNumber(int number) {
        this.number = number;
    }

    // Getter method for name
    public String getName() {
        return name;
    }

    // Setter method for name
    public void setName(String name) {
        this.name = name;
    }

    // Method to print details
    public void printDetails() {
        System.out.println("Name: " + name);
        System.out.println("Number: " + number);
    }
}
```