FULL STACK DEVELOPMENT – WORKSHEET 3

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

Q2. Which of these cannot be used for a variable name in Java?
A. identifier & keyword

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

Q4.Which one is a valid declaration of a boolean?
C. boolean b3 = false;

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

Q6.All the variables of the interface should be?
C. public, static and final

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

Q8.How many arguments can be passed to main()?
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]); } }
C. This

Q10.What is the value of “d” in the following Java code snippet? double d = Math.round (2.5 + Math.random());
B. 3

Q11.Which of these methods is a rounding function of Math class?
D. all of the mentioned

Q12. Standard output variable ‘out’ is defined in which class?
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); } } }
B. 8

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

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

C. Runtime Error

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);
}
}
```

D. 30

Q18. Write Syntax to create/define java methods.

```
// Method with no parameters and no return value (void)
public void printMessage() {
    System.out.println("Hello, World!");
}

// Method with parameters and a return value (int)
public int add(int a, int b) {
    int sum = a + b;
    return sum;
}

// Method with a parameter and no return value (void)
private void greet(String name) {
    System.out.println("Hello, " + name + "!");
}
```

Q19. Write a java program following instructions
A. Make a class Addition
a. initialize sum as 0
b. make addTwoInt method taking two int parameters a,b. make sum = a+b. Return Sum

```
class Addition {
    int sum = 0;

    int addTwoInt(int a, int b) {
        sum = a + b;
        return sum;
    }

    public static void main(String[] args) {
        Addition addition = new Addition();
        int result = addition.addTwoInt(5, 7);
        System.out.println("Sum: " + 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

```
class MethodCall {
    public static void main(String[] args) {
        Addition addition = new Addition(); // Create an object of the Addition class
        int result = addition.addTwoInt(5, 7); // Call the addTwoInt method using the
object
        System.out.println("Sum: " + result); // Print the sum
    }
}
```

Write a java program following the 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

```
class Example {
    // Instance variables
    private int number;
    private String name;
    // Accessor methods (getters)
    public int getNumber() {
        return number;
    }

    public String getName() {
        return name;
    }

    // Mutator methods (setters)
    public void setNumber(int number) {
        this.number = number;
    }

    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 the Example class
        Example example = new Example();

        // Set values using mutator methods
        example.setName("John Doe");
        example.setNumber(42);

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

Q20. Write a java program following the instructions

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

```java
class Example {
    // Instance variables
    private int number;
    private String name;

    // Accessor methods (getters)
    public int getNumber() {
        return number;
    }

    public String getName() {
        return name;
    }

    // Mutator methods (setters)
    public void setNumber(int number) {
        this.number = number;
    }

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

        // Set values using mutator methods
        example.setName("Your name");
        example.setNumber(123);

        // Call printDetails method using the instance
        example.printDetails();
    }
}
```