`FULL STACK DEVELOPMENT – WORKSHEET 3

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

A. Object-oriented
B. Use of pointers
C. Portable
D. Dynamic and Extensible

Ans:- Use of pointers

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

A. identifier & keyword
B. identifier
C. keyword
D. none of the mentioned

Ans:- keyword

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

A. ArrayList
B. Abstract class
C. Object class
D. String

Ans:- Object class

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

A. boolean b1 = 1;
B. boolean b2 = 'false';
C. boolean b3 = false;
D. boolean b4 = 'true'

Ans:- boolean b3 = false;

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

A. protected
B. private
C. public
D. default

Ans:- default

Q6.All the variables of interface should be?

A. **default and final**
B. **default and static**
C. **public, static and final**
D. **protect, static and final**

Ans:- public, static and final

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

A. **Array**
B. **Stack**
C. **String**
D. **Integer**

Ans:- string

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

A. **Infinite**
B. **Only 1**
C. **System Dependent**
D. **None of the mentioned**

Ans:- Infinite

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]);
   }
}
```

A. java
B. Output
C. This
D. Is

Ans:- This

Q10.What is the value of “d” in the following Java code snippet?

```
double d = Math.round ( 2.5 + Math.random() );
```

A. 2
B. 3
C. 4
D. 2.5

Ans:- 3

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

A. max()
B. min()
C. abs()
D. all of the mentioned

Ans:- all of the mentioned

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

A. Void
B. Process
C. Runtime
D. System

Ans:- 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);
        }
    }
}
```

A. 9
B. 8
C. Compilation error
D. Runtime error

Ans:- Compilation error

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
B. static method
C. private method
D. finalize method

Ans:- static method

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

A. Parameters data type
B. Number of parameters
C. Return type of method
D. All of the mentioned

Ans:- 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);
        }
    }
```

A. 1
B. 2
C. Runtime Error
D. Compilation Error

Ans:- 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);
```

```
        }
    }
```

A. 0
B. 1
C. 25
D. 30

Ans:- 30

Q18. Write Syntax to create/define java methods.

Ans:-

```
AccessModifier ReturnType
    methodName(ParameterList) {
  // Method Body
  // Statements to be executed
}
```

AccessModifier: It specifies the access level of a method, whether it can be accessed outside the class or not.

ReturnType: It is the data type of the value returned by the method after execution. It can be a primitive data type, object, or void if the method does not return any value.

methodName: It is the name given to the method.

ParameterList: It contains the data types and the names of the parameters passed into the method. It can be empty if there are no parameters in the method.

Method Body: It contains the block of code that makes up the method and is executed when the method is called.

Example:

```
public int AddNumbers(int a, int b) {
  int sum = a + b;
  return sum;
    }
```

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

Ans:-

```
public class Addition {
  int sum = 0;

  // Method to Add Two Integers
  public int addTwoInt(int a, int b) {
    sum = a + b;
    return sum;
  }
}
```

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:-

```
// MethodCall Class
public class MethodCall {
 // Main Method
 public static void main(String[] args) {
  // Create Instance of Addition Class
  Addition obj = new Addition();

  // Call addTwoInt Method
  int result = obj.addTwoInt(10, 20);

  // Print Sum
  System.out.println("Sum of Two Integers: " + result);
```

```
 }
}
```

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:-

```
/ Example Class
class Example {
   private int number;
   private String name;

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

  public String getName() {
  return name;
  }

 // Mutator Methods
 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);
  }
}
```

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 {
// Demo Class
  public static void main(String[] args) {
    // Create Instance of Example Class
    Example example = new Example();

    // Set Number and Name
    example.setNumber(123);
    example.setName("Tejashwini D");

    // Call Print Details Method
    example.printDetails();
  }
}
```

OUTPUT

```
Name: Tejashwini D
Number: 123
```