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:B

Pointer is not a java feature.

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:c

Q3.Which of the following is a superclass of every class in Java? A. ArrayList B. Abstract class C. Object class D. String

Ans:c

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: C

"Boolean" is a "data type" that can either be "True or False".

Q5. Which is the modifier when there is none mentioned explicitly? A. protected B. private C. public D. default

Ans: D

Default is the access modifier when none is defined explicitly.

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:c

Explanation: Variables of an interface are public, static and final by default because the interfaces cannot be instantiated, final ensures the value assigned cannot be changed with the implementing class and public for it to be accessible by all the implementing classes.

Q7.Which of these data types is used to store command line arguments? A. Array B. Stack C. String D. Integer

Ans: C

All command Line arguments are passed as a string.

How many arguments can be passed to main()? A. Infinite B. Only 1 C. System Dependent D. None of the mentioned

Ans A

Infinite number of arguments can be passed to main()

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 : A

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:B

The Math.random() method returns a number greater than or equal to 0 and less than 1 . Since we can then be sure that the sum of that number and 2.5 will be greater than or equal to 2.5 and less than 3.5, we can be sure that Math.round() will round that number to 3. So Option B is the answer.

Q11.Which of these methods is a rounding function of Math class? A. max() B. min() C. abs() D. all of the mentioned

Ans: D

Q12. Standard output variable ‘out’ is defined in which class? A. Void B. Process C. Runtime D. System

Ans:D

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:C

Two variables with the same name can’t be created in a class. output:

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: B

If a static method is present in the program then it will be executed first, then main will be executed.

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: D

Return type of method, Number of parameters and Parameters data type can be used to differentiate two or more methods having same name.

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: B

Operator short circuit and, &&, skips evaluating right hand operand if left hand operand is false thus division by zero in if condition does not give an 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);

 }

 }
```

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

Ans:c

Q18. Write Syntax to create/define java methods.

A **method** is a block of code which only runs when it is called.

You can pass data, known as parameters, into a method.

Methods are used to perform certain actions, and they are also known as **functions**.

To reuse code: define the code once, and use it many times.

Create a Method

A method must be declared within a class. It is defined with the name of the method, followed by parentheses **()**. Java provides some pre-defined methods, such as System.out.println(), but you can also create your own methods to perform certain actions:

Create a method inside Main:

```
Public class Main{

Static void myMethod

//code to execute

}

}
```

Example Explained

- myMethod() is the name of the method

- static means that the method belongs to the Main class and not an object of the Main class. You will learn more about objects and how to access methods through objects later in this tutorial.
- void means that this method does not have a return value. You will learn more about return values later in this chapter

Call a Method

To call a method in Java, write the method's name followed by two parentheses **()** and a semicolon**;**

In the following example, myMethod() is used to print a text (the action), when it is called:

Example

```
Public class Main{

Static Void myMethd(){

System.out.println(“Hello World”);

}

Public static void main(String[] args){

myMethod();

}

}
```

```
// Outputs "I just got executed!"
```

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

```
import java.util.Scanner;
class Addition
{
public static void main(String args[])
{
int x, y, sum =0;
Scanner sc = new Scanner(System.in);
x = sc.nextInt();
y = sc.nextInt();
sum = addTwo(x, y);
System.out.println("The sum of numbers " + x + " and " + y + " is: " +sum );
}
public static int addTwo(int a, int b)
{
int 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

```
import java.util.Scanner;

public class Calculate
{
    private int a;
    private int b;
    private int sum;
    private int diff;

    public void inputdata() {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter first number: ");
        a = in.nextInt();
        System.out.print("Enter second number: ");
        b = in.nextInt();
    }

    public void Addition() {
        sum = a + b;
    }

    public void outputdata() {
  System.out.println("Sum = " + sum);

    }

    public static void main(String args[]) {
        Addition obj = new Addition();
        obj.inputdata();
        obj.addition();
        obj.outputdata();
    }
}
```

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

```
public class Employee {
    private int number;
    public int getNumber() {
        return number;
    }
    public void setNumber(int newNumber) {
        number = newNumber;
    }
}
```

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

```
class Demo{
 int rollno;
 String name;
 void insertRecord(int r, String n){
  rollno=r;
  name=n;
 }
 void displayInformation(){System.out.println(rollno+" "+name);}
}
class TestDemo{
 public static void main(String args[]){
  Student s1=new Demo();
  Student s2=new Demo();
  s1.insertRecord(123,"Ajeet");
  s2.insertRecord(122,"Aryan");
  s1.displayInformation();
  s2.displayInformation();
 }
}
```