FULL STACK DEVELOPMENT – WORKSHEET 4

Q1.Write in brief about OOPS Concept in java with Examples. (In your own words)

Ans:

Object-oriented programming (oops) is afundamental programming paradigm based on the concept of object.

1. Class: It is a blue print , It defines the properties and behaviours that object of that class will have.

```
public class Person {
    String name;
    int age
    public int getAge() {
        return age;
    }
    public void speak() {
        System.out.println(name + " says hello!");
    }
}
```

2.Object : It is an instance of a class. An entity that has state and behaviour.

```
Person person1 = new Person();
person1.name = "Alice";
person1.age = 30;

Person person2 = new Person();
person2.name = "Bob";
person2.age = 25;
```

3. Encapsulation: it is a concept of binding data and methods that operate on that data into a single unit.

```
public class Person {
    private String name;
    private int age;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

4. Inheritance: It allows you to create a new class by inheriting properties and behaviours from an existing class.

```
public class Student extends Person {
    private String studentId;
    public String getStudentId() {
        return studentId;
    }
    public void setStudentId(String studentId) {
        this.studentId = studentId;
    }
}
```

5.Ploymrphisim : It is the ability of an object may task.It performs same task in different forms.

```
public class Animal {
    public void makeSound() {
        System.out.println("Some generic animal sound.");
    }
}
public class Dog extends Animal {
    public void makeSound() {
        System.out.println("Woof!");
    }
}
```

```
public class Cat extends Animal {

   public void makeSound() {

      System.out.println("Meow!");

   }

}
```

6. Abstraction: Hiding the functionalities.There are two ways to achieve abstraction:

Using abstract keyword and by using abstract interface.

Q2.Write simple programs(wherever applicable) for every example given in Answer 2.

Multiple Choice Questions

Q1. Which of the following is used to make an Abstract class?

A. Making at least one member function as pure virtual function
B. Making at least one member function as virtual function
C. Declaring as Abstract class using virtual keyword
D. Declaring as Abstract class using static keyword

Ans: A. Making at least one member function as pure virtual function

```
Class Main{
        Public:
                Virtual void Mainfunc()=0
};
```

Q2. Which of the following is true about interfaces in java.

1) An interface can contain the following type of members.
....public, static, final fields (i.e., constants)
....default and static methods with bodies
2) An instance of the interface can be created.
3) A class can implement multiple interfaces.
4) Many classes can implement the same interface.

A. 1, 3 and 4
B. 1, 2 and 4
C. 2, 3 and 4
D. 1,2,3 and 4

Ans: A.1, 3 and 4

Q3. When does method overloading is determined?

A. At run time
B. At compile time
C. At coding time
D. At execution time

Ans: B. At compile time

Q4.What is the number of parameters that a default constructor requires?

A. 0
B. 1
C. 2
D. 3

Ans: A. 0

Q5.To access data members of a class, which of the following is used?

A. Dot Operator
B. Arrow Operator
C. A and B both as required
D. Direct call

Ans: A. Dot Operator

Q6.Objects are the variables of the type_______?

A. String
B. Boolean
C. Class
D. All data types can be included

Ans: C. Class

Q7.A non-member function cannot access which data of the class?

A. Private data
B. Public data
C. Protected data
D. All of the above

Ans: A. Private data

Q8. Predict the output of following Java program

```
 class Test {
   int i;
 }
 class Main {
   public static void main(String args[]) {
       Test t = new Test();
       System.out.println(t.i);
            }
}
```

A. garbage value
B. 0

C. compiler error
D. runtime Error

Ans: B. 0

Q9.Which of the following is/are true about packages in Java?

1) Every class is part of some package.

2) All classes in a file are part of the same package.

3) If no package is specified, the classes in the filego into a special unnamed package

4) If no package is specified, a new package is created with folder name of class and the class is put in this package.

A. Only 1, 2 and 3
B. Only 1, 2 and 4
C. Only 4
D. Only 1, 3 and 4

Ans: A. Only 1, 2 and 3

For Q10 to Q25 find output with explanation.

Q10.Predict the Output of following Java Program.

```
class Base {
    public void show() {
       System.out.println("Base::show() called");
    }
}
class Derived extends Base {
    public void show() {
       System.out.println("Derived::show() called");
    }
}
public class Main {
    public static void main(String[] args) {
        Base b = new Derived();
        b.show();
    }
}
```

Ans: Derived::show() called

Object of derived class is created and assigned to reference variable of base class. Overridden method of derived class is getting executed.

Q11. What is the output of the below Java program?

```
class Base {
    final public void show() {
       System.out.println("Base::show() called");
    }
}
class Derived extends Base {
    public void show() {
       System.out.println("Derived::show() called");
    }
}
class Main {
    public static void main(String[] args) {
        Base b = new Derived();;
        b.show();
    }
}
```

Ans: Base::show() called

As final keyword is used in the Base class method, it cannot be overridden in the derived class.

Q12.Find output of the program.

```
class Base {
    public static void show() {
       System.out.println("Base::show() called");
    }
}
class Derived extends Base {
    public static void show() {
       System.out.println("Derived::show() called");
    }
}
class Main {
    public static void main(String[] args) {
        Base b = new Derived();
        b.show();
    }
}
```

Ans: Derived::show() called

Static method show is overridden in the derived class, so when object of base class is called the derived class’s method get executed.

Q13.What is the output of the following program?

```
class Derived
{
    public void getDetails()
    {
        System.out.printf("Derived class ");
    }
}
public class Test extends Derived
{
    public void getDetails()
    {
        System.out.printf("Test class ");
        super.getDetails();
    }
    public static void main(String[] args)
    {
        Derived obj = new Test();
        obj.getDetails();
    }
}
```

Ans: Test class Derived class

First when object of Derived class is called, the method from Test class gets executed as it is overridden.
Test class’s method calls the supperclass’s method using keyword super.So it prints Derived class.

Q14. What is the output of the following program?

```
class Derived
{
    public void getDetails(String temp)
    {
        System.out.println("Derived class " + temp);
    }
}
public class Test extends Derived
{
    public int getDetails(String temp)
    {
        System.out.println("Test class " + temp);
        return 0;
    }
    public static void main(String[] args)
    {
        Test obj = new Test();
        obj.getDetails("Name");
    }
}
```

Ans: Compilation Error
The return type is incompatible with Derived.getDetails(String)

Q15.What will be the output of the following Java program?

```
    class test
{
          public static int y = 0;
}
class HasStatic
{
          private static int x = 100;

          public static void main(String[] args)
          {
                    HasStatic hs1 = new HasStatic();
                    hs1.x++;
                    HasStatic hs2 = new HasStatic();
                    hs2.x++;
                    hs1 = new HasStatic();
                    hs1.x++; HasStatic.x++;
                    System.out.println("Adding to 100, x = " + x);test
                    t1 = new test();
                    t1.y++;
                    test t2 = new test();
                    t2.y++;
                    t1 = new test();
                    t1.y++;
                    System.out.print("Adding to 0, ");
                    System.out.println("y = " + t1.y + " " + t2.y + " " + test.y);
          }
}
```

Ans: Adding to 100, x = 104
Adding to 0, y = 3 3 3

Q16.Predict the output

```
class San
{
 public void m1 (int i,float f)
 {
  System.out.println(" int float method");
 }
 public void m1(float f,int i);
  {
  System.out.println("float int method");
  }
  public static void main(String[]args)
  {
    San s=new San();
        s.m1(20,20);
  }
}
```

Ans: compilation error
The method m1(int, float) is ambiguous for the type San

Q17.What is the output of the following program?

```
public class Test
{
    public static void main(String[] args)
    {
        int temp = null;
        Integer data = null;
        System.out.println(temp + " " + data);
    }
}
```

Ans: CompilationError
Cannot convert type int to null

Q18.Find output

```
class Test {
    protected int x, y;
}

class Main {
    public static void main(String args[]) {
        Test t = new Test();
        System.out.println(t.x + " " + t.y);
    }
}
```

Ans: 0 0

Q19.Find output

```
// filename: Test2.java
class Test1 {
          Test1(int x)
          {
                    System.out.println("Constructor called " + x);
          }
}
class Test2 {
          Test1 t1 = new Test1(10);
          Test2(int i) { t1 = new Test1(i); }
          public static void main(String[] args)
          {
                Test2 t2 = new Test2(5);
        }
}
```

Ans: Constructor called 10
        Constructor called 5

Q20.What will be the output of the following Java program?

```
class Main
{
 public static void main(String[] args)
  {
  int []x[] = {{1,2}, {3,4,5}, {6,7,8,9}};
  int [][]y = x;
  System.out.println(y[2][1]);
  }
}
```

Ans: 7

Q21.What will be the output of the following Java program?

```
class A
{
    int i;
    public void display()
    {
        System.out.println(i);
    }
}
class B extends A
{
    int j;
    public void display()
    {
        System.out.println(j);
    }
}
class Dynamic_dispatch
{
    public static void main(String args[])
    {
        B obj2 = new B();
        obj2.i = 1;
        obj2.j = 2;
        A r;
        r = obj2;
        r.display();
    }
}
```

Ans: 2

Q22. What will be the output of the following Java code?

```
class A
{
    int i;
    void display()
    {
        System.out.println(i);
    }
}

class B extends A
{
    int j;
    void display()
    {
        System.out.println(j);
    }
}
class method_overriding
{
    public static void main(String args[])
    {
        B obj = new B();
        obj.i=1;
        obj.j=2;
        obj.display();
    }
}
```

Ans: 2
It is printing the value of j from the obj and the method overriding feature ensures that the display method of class B is called

Q23.What will be the output of the following Java code?

```
class A
{
    public int i;
    protected int j;
}
class B extends A
{
    int j;
    void display()
    {
        super.j = 3; System.out.println(i
        + " " + j);
    }
}
class Output
{
    public static void main(String args[])
    {
        B obj = new B();
        obj.i=1;
```

```
            obj.j=2;
            obj.display();
        }
```

Ans: 1 2

Q24.What will be the output of the following Java program?

```
    class A
    {
        public int i;
        public int j;

        A()
        {
            i = 1;
            j = 2;
           }
    }
    class B extends A
    {
        int a;
        B()
        {
            super();
        }
    }
    class super_use
    {
        public static void main(String args[])
        {
            B obj = new B();
            System.out.println(obj.i + " " + obj.j)
        }
   }
```

Ans: 1 2

Q 25. Find the output of the following program.

```
class Test
{
    int a = 1;
    int b = 2;

    Test func(Test obj)
    {
        Test obj3 = new Test();
        obj3 = obj;
        obj3.a = obj.a++ + ++obj.b;
        obj.b = obj.b;
        return obj3;
    }
    public static void main(String[] args)
    {
        Test obj1 = new Test();
        Test obj2 = obj1.func(obj1);

        System.out.println("obj1.a = " + obj1.a + " obj1.b = " + obj1.b);
        System.out.println("obj2.a = " + obj2.a + "  obj1.b = " + obj2.b);

    }
 }
```

Ans: obj1.a = 4 obj1.b = 3
      obj2.a = 4 obj1.b = 3