FULL STACK DEVELOPMENT – WORKSHEET 4

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

Java is a Object Oriented Programming language. It has 4 OOPS features,

1) **Abstraction** : In JAVA Abstraction is used to hide complex and unnecessary details, shows only important part. Abstraction is achieved by using **Interfaces and Abstract classes.**

   For example, In a car only driver can see only outer parts of a car like steering, Gearbox, speedometer while driver need not to worry about functionality of a main part which is engine. Thus Abstraction hides functionality.

2) **Encapsulation** : In JAVA Encapsulation is used to hide data to keep it safe from secure from misusing it. A **class members are declared as private** so that they can be accessed outside of a class, through its public methods only. **Encapsulation hides Data**.

   For example, while driving a car, driver has no access to engine which is a private part of car whereas driver can access public parts like breaks to slow or speedup the car.

3) **Inheritance :** Inheritance is the mechanism where a class can inherit properties of a parent class by extending it so that old functionality can be reused and new functionalities can be added to a new class.

   For example, launching of a new version of a car, manufacturing company inherits old version of a car, inheriting old features as well as adds new features to new version**.**

4) **Polymorphism** : It means many forms where different actions are performed through one interface. **Polymorphism is achieved through overloading and overriding.**

   For example, if a dog smells a cat it will bark but if it smells food then it will salivates. So smelling is a function which acts differently depends upon type of arguments passed to it.

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

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

Q3. When does method overloading is determined?

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

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

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

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

Q6. Objects are the variables of the type____?

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

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

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

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 file go 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

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();
    }
}
```

Output : Derived::show() called

(Explaination : Base b = new Derived();

Using this statement we are creating an object of type Derived class and assigning it to a reference variable of type Base class. So here 'b.Show()' is refering to overridden method in a Derived class)

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();
    }
}
```

Answer : There will be no output as it will show a compile time error, “final methods can not be overriden”.

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();
    }
}
```

Output : output will be “Base::show() called”.

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();
    }
}
```

Output: Test class 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");
   }
}
```

Output: Derived class Name (Explanation : object obj can access method of derived class and its own class method. Whenever a method is called through an object, it matches with the method definition and no. as well as type of parameters passed to it. According to that, regarding method will be called. Here, string is passed as an argument hence method of Derived class is called.)

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);
```

```
        }
}
```

Output : 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);
  }
}
```

Answer : It will show a compile time error, “The method m1 is ambiguous for the type San class.” (Explanation : compiler will get confused, which method to call because both methods have matching parameters)

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);
   }
}
```

Answer : It will show a compile time error, “Can not convert null to type integer”

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);
    }
}
```

Output : 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);
        }
}
```

Output : 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]);
   }
 }
```

Output : 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();
    }
}
```

Output : 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();
    }
}
```

Output : 2

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();
    }
}
```

Output : 1 2 (Explanation : Here class B exteds class A. Class A has Protected variable 'j'. To access this variable we have to refer it as super.j or A.j. As class B also has Varible j, that's why value of i and j are printed as 1 and 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)
    }
}
```

Output : 1 2 (Explanation : Here B extends class A. As Constructor Of B is called, bydefault call goes to Super constructor. Hence values printed are 1 and 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);
```

```
    }
}
```

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