FULL STACK DEVELOPMENT – WORKSHEET 4

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

Ans –

Java is an object-oriented programming (OOP) language that uses classes and is based on the idea of objects. The goal of OOP principles is to increase the readability and re - usability of code. There are four principles of object-oriented programming:

• **Abstraction**- With abstraction, consumers can only see the information that matters. It can be achieved by making abstract classes and interfaces. **Example- Coffee machines**.

• **Encapsulation**- Data security is aided by encapsulation, which lets us shield a class's data from system-wide access by acting as a capsule to protect a class's internal information. Making the fields private and accessing them through getter and setter methods, allows us to build encapsulation in Java. **Example- Bank accounts**.

• **Inheritance** makes it possible to create a child class that inherits the fields and methods of the parent class. **Example – Dogs and cats inheriting an Animal class**.

• **Polymorphism**- Method overloading and method overriding are two ways that polymorphism can be implemented in Java. When a class contains multiple methods with the same name, this is known as method overloading. They can be distinguished by their parameters. When a child class overrides a parent class's method, this is known as method overriding. Example - A individual who can has different relationships with is a real-world example of polymorphism like , a man is a father to his child, a husband to his partner, and a worker to his employer.

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

Ans- In code

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 - The way to create an abstract class is to turn at least one member function into a pure virtual function**. Hence, the correct option is (a).**

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 - Hence, the correct option is (a).

Q3. When does method overloading is determined?

A. At run time

B. At compile time

C. At coding time

D. At execution time

Ans - Method Overloading is determined at compile time. **Hence, the correct option is (b).**

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

A. 0

B. 1

C. 2

D. 3

Ans - A default constructor does not require any parameters. **Hence, the correct option is (a).**

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 - Dot operator is used to access the members with help of object of class. **Hence, the correct option is (a).**

Q6. Objects are the variables of the type ____?

A. String

B. Boolean

C. Class

D. All data types can be included

Ans - Once the class has been established, any number of objects that fall under that class can be produced. **Hence, the correct option is (c).**

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 non- member function cannot access private data of the class. **Hence, the correct option is (a).**

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 - The output will be 0 as there is no value assigned to int i. **Hence, the correct option is (b).**

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

Ans - **Hence, the correct option is (a).**

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 - The output will be "Derived::show() called" as there is a base class – Base and a derived class- Derived. In the main method, an object is created which is a reference that points to a Derived class subject. As a result, derived fun() is invoked and run-time polymorphism takes place.

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();
}
}
```

Output - The program will throw a compile error as final methods cannot be overridden.

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 - . The output will be **"Base::show() called"** as the methods are made static.

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 - The output will be **Test class Derived class** as in the test class, print statement will execute first, then due to super keyword referring to the upper method, derived class print statement will execute second.

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 - The program **will throw a compile error** as there is an int data type associated with the getDetails method that overrides the Derived.getDetails method.

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 - The output will be :

Adding to 100, x = 104 // because x has been incremented 4 times.

Adding to 0, y = 3 3 3 // because y has been incremented 3 times but printed only one time. Because Static variables are global variables.

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);
} }
```

Output - The program will **throw a compile error** as there is an ambiguity for compiler to call which m1 method and there is semi colon in place for method body in second m1 method.

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);
} }
```

Output - The program will **throw a compile error** as Null values cannot be assigned to primitive data types.

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 - The output will be 0 0 as the default constructors initialize variables as 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 - The output will be:

Constructor called 10

Constructor called 5

Because, inside the class Test2, Test1 object is created and the object for class Test2 is created in main class.

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 - The output will be **7** because the first element in array 2 is 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 - . The output will be **2** as r is made a reference of type A, r.display prints the value.

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 - The output will be **2** as b is made as a reference for class B and the value of j is 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 - The output will be 1 2 because when no specifier is declared, the member of class B will be called by default because both classes A and B include members with the same name, j. 1 2 will be printed because i = 1 and j = 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 - The output will be 1 2 because the super keyword calls the constructor of class A by constructor of class B, where the constructor of A initializes i = 1 & j= 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 - The output will be:

obj1.a = 4 obj1.b = 3

obj2.a = 4 obj1.b = 3

as the value of a and b is incremented and obj1 &obj2 refer to same memory address.