Worksheet – 4

Q1. Write in brief about OOPS Concept in java with Examples.

Ans. Java, renowned for its simplicity, reliability and portability has been a cornerstone in the programming landscape for decades. With OOPs as its guiding principle, Java empowers developers to think inn terms of real-world objects, encapsulating data and behaviour into reusable building blocks. From the tiniest applications to large-scale enterprise systems, OOPs in Java paves the way for modular, extensible, and robust software solutions that transcend traditional procedural programming. So, fasten your seatbelts as we embark on a thrilling journey into the principles, concepts, and techniques that make OOPs in Java a game-changer in the world of programming.

OOPs in java is to improve code readability and reusability by defining a Java program efficiently.

The main principles of object-oriented programming are abstraction, encapsulation, inheritance and polymorphism. These concepts aim to implement real-world entities in programs.

Multiple Choice Questions

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

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

Q2. Which of the following is true about interfaces in Java?

Ans. 1, 3 and 4

Q3. When does method overloading is determined?

Ans. At Compile time

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

Ans. 0

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

Ans. A and B both are required

Q6. Objects are the variables of the type?

Ans. Class

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

Ans. Private data

Q8. Predict the output of the following java program?

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

}

Ans. 0

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

Ans. 1, 2 and 3

For Q10 to Q25 Find Output.

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. Output :

```
Derived::show() called
```

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. Output

Compile error

Q12. Find output of the given 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. Output

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();
}
}
```

Ans. 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");
}
}
```

Ans. Output

Compilation error

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. 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");
WORKSHEET
}
public static void main(String[]args)
{
San s=new San();
s.m1(20,20);
}
}
```

Ans. Output

Compile time error

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. Output

Compilation error due to temp

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. 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);
}
}
```

Ans. Output

Constructor called 10

Constructor called 5

Q20. What will be the output of the following j ava 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. Output

7

Q21. What will be the output of the following j ava 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. Output

2

Q22. What will be the output of the following j ava 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. Output

2

Q23. What will be the output of the following j ava 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. Output

1 2

Q24. What will be the output of the following j ava 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. Output

```
1 2
```

Q25. 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. Output

obj1.a = 4 obj1.b = 3

obj2.a = 4 obj2.b = 3