FULL STACK DEVELOPMENT – WORKSHEET 4 (Answers)

Q1. Write in brief about OOPS Concept in java with Examples.
Answer: Oops is a fundamental programming paradigm based on the concept of "Objects". An object is an entity in the real world that can be directly identified. They consist of methods and properties to make a particular type of data useful.

Pillars of OOPS are

1. **Object**
2. **Class: is the collection of objects and is also a logical entity**
3. **Inheritance: Acquiring all the properties and behaviors of a parent class**
4. **Polymorphism: Performing one task in different ways. It is achieved through method overloading and method overriding**
5. **Abstraction: Hiding the internal details and showing only the functionality**
6. **Encapsulation: Wrapping code and data together into a single unit. This is achieved through getter and setter methods.**

Multiple Choice Questions

Q1. Which of the following is used to make an Abstract class?
Answer: B. Making at least one member function as virtual function

Q2. Which of the following is true about interfaces in java.
Answer: A. 1, 3 and 4

Q3. When does method overloading is determined?
Answer: B. At compile time

Q4. What is the number of parameters that a default constructor requires?
Answer: A. 0

Q5. To access data members of a class, which of the following is used?
Answer: A. Dot Operator

Q6. Objects are the variables of the type ____?
Answer: C. Class

Q7. A non-member function cannot access which data of the class?
Answer: 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);
        }
}
```

Answer: B. 0

Q9. Which of the following is/are true about packages in Java?
Answer: A. Only 1, 2 and 3

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();
}}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q10.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Q10
Derived::show() called
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

Because the derived class overrides the base 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:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q11.java
Q11.java:8: error: show() in Derived cannot override show() in Base
    public void show() {
                ^
  overridden method is final
1 error
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

Once declared as final, the variable or method cannot be re-declared or 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();
	}
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q12.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Q12
Base::show() called
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

Because the object is created for the base class, not the derived class hence the base class method is 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();
	}
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q13.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Q13
Test class Derived class
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

The derived method is called first which in turn is calling the same method from its Super 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");
	}
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q14.java
Q14.java:10: error: getDetails(String) in Q14 cannot override getDetails(String) in Derived
    public int getDetails(String temp)
               ^
  return type int is not compatible with void
1 error
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

For method overriding, the return type must be same

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);
        }
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q15.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Q15
Adding to 100, x = 104
Adding to 0, y = 3 3 3
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

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:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q16.java
Q16.java:7: error: missing method body, or declare abstract
    public void m1(float f,int i);
                ^
Q16.java:14: error: reference to m1 is ambiguous
        s.m1(20,20);
         ^
  both method m1(int,float) in Q16 and method m1(float,int) in Q16 match
2 errors
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

- Second m1 method has a semi colon and hence the control is unable to enter the method body
- Both the methods m1 match with return type as well as the parameters hence the ambiguity

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:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q17.java
Q17.java:4: error: incompatible types: <null> cannot be converted to int
        int temp = null;
                   ^
1 error
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

Int variable cannot be null

Auto-boxing is happening through the wrapper class in Integer temp variable

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);
	}
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q18.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Q18
0 0
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

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);
	}
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac test2.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Test2
Constructor called 10
Constructor called 5
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

Test 1 constructor is initiated first with argument 10 and then Test 2 constructor is initiated with passing 5 as argument which inturn calls Test 1 with the same integer 5 as argument hence argument 10 is printed first and then argument 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]);
	}
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q20.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Q20
7
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

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();
        }
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q21.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Q21
2
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

Object reference obj2 is assigned to object r as reassignment. And r is calling the display method of overriding derived class

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();
        }
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q22.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Q22
2
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

Deriver class is called and hence value 2 is passed and same is displayed

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();
        }
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q23.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Q23
1 2
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

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)
        }
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q24.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Q24
1 2
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```

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);
        }
}
```

Answer:

```
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> javac Q25.java
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4> java Q25
obj1.a = 4 obj1.b = 3
obj2.a = 4 obj1.b = 3
PS C:\Users\suhai\Documents\FlipRoboInternship\Worksheet-4>
```