*Object Orientation Assignment-1*

1. What is the output of the following program?

package main;

class Parent {

public void display() {

System.out.println("Parent");

}

}

class Child extends Parent {

public void display() {

System.out.println("Derived");

}

}

class Main{

public static void doDisplay( Parent o ) {

o.display();

}

public static void main(String[] args) {

Parent x = new Parent();

Parent y = new Child();

Child z = new Child();

doDisplay(x);

doDisplay(y);

doDisplay(z);

}

}

*Output:*

*Parent*

*Derived*

*Derived*

2. What is the output of following program?

class Coordinate {

int x, y;

public Coordinate(int x, int y) {

this.x = x;

this.y = y;

}

}

public class Main {

public static void main(String args[]) {

Coordinate p = new Coordinate();

System.out.println("x = " + p.x + ", y = " + p.y);

}

}

*Output:*

*Error*

*Error will occur because we are not passing any argument in the constructor while declaring the object of class Coordinate. However, we have declared the parameters in constructors so we should pass arguments.*

3. What is the output of following program?

public class Test {

int x = 2;

Test(int i) {

x = i;

}

public static void main(String[] args) {

Test t = new Test(5);

System.out.println("x = " + t.x);

}

}

*Output:*

*X = 5*

4. What is the output of following program?

class SuperClass {

protected void foo() {}

}

class Child extends SuperClass {

void foo() {}

}

public class Test {

public static void main(String args[]) {

Child child = new Child();

child.foo();

}

}

*Output:*

*Error*

*It will show error because foo() method in sub-class overrides foo() method of SuperClass with different access modifier. Foo() of SuperClass has protected access modifier and foo() of child class has default access modifier. As per the rule of method overriding the access modifier of both the methods should be same.*

5. What is the output of the following program?

public class Test {

public static void test(String s)

{

System.out.println("In String");

}

public static void test(Object o) {

System.out.println("In Object");

}

public static void main(String args[]) {

test(null);

}

}

*Output:*

*In String*

*Object Orientation Assignment-2*

1. How do you implement multiple inheritance in java?

1. Multiple inheritance can only be possible through interface because in case of interface there is no ambiguity means the diamond problem does not exist as interface methods are implemented by child classes only.

2. How do you restrict a member of a class from inheriting to it’s sub classes.?

2. By declaring member of parent class private.

3. Are constructors and initializers also inherited to sub classes?

3. A subclass inherits all the *members (*fields, methods, and nested classes) from its superclass. Constructors are not members, so they are not inherited by subclasses, but the constructor of the superclass can be invoked from the subclass.

4. Are static members inherited to sub classes?

4. Yes static members also inherited to sub classes and we can use them directly in the sub classes.

5. Can we use a field or a method declared without access modifiers outside the package?

5. No, we cannot use field or a method declared without access modifiers outside the package.

6. Can a top level class be private or protected?

6. Access level modifiers determine whether other classes can use a particular field or invoke a particular method. There are two levels of access control:

* At the top level—public, or *package-private* (no explicit modifier).
* At the member level—public, private, protected, or *package-private (*no explicit modifier).