1. **The output of the program will be:**

Person

Student

Person

java.lang.Object

1. If we change "o1" to "o" in the if statement, the program would crash because the "o1" object is an instance of "Object", not "GraduateStudent". When the explicit casting is attempted, a ClassCastException would be thrown because it is not possible to cast an instance of "Object" to "GraduateStudent".
2. To avoid this crash, we can check the type of an object before attempting to cast it to another type. we can use the "instanceof" operator or the "getClass" method to check the actual type of an object. If the object is not of the expected type, we can handle the error by throwing an exception or returning a default value. For example:

if (o instanceof GraduateStudent) {

GraduateStudent gs = (GraduateStudent) o;

// do something with gs

} else {

// handle error or return default value

}

Or we can Using a try-catch block to catch the ClassCastException that may be thrown when attempting to cast an object to another type.

try {

GraduateStudent gs = (GraduateStudent) o;

// do something with gs

} catch (ClassCastException e) {

// handle error or return default value

}

public class DynamicBindingDemo2 {

public static void main(String[] args) {

Object o = new GraduateStudent();

if (o instanceof GraduateStudent) {

GraduateStudent gs = (GraduateStudent) o;

System.out.println("The object is a GraduateStudent");

} else {

System.out.println("The object is not a GraduateStudent");

}

m(o);

m(new Student());

m(new Person());

m(new Object());

}

public static void m(Object x) {

System.out.println(x.toString());

}

}

class GraduateStudent extends Student {

}

class Student extends Person {

public String toString() {

return "Student";

}

}

class Person extends Object {

public String toString() {

return "Person";

}

}

**Program output Screen Shot**