

# OOP Lab Report - 02

## **Submitted to:**

Teacher's Name : Md. Ismail

Designation : Lecturer

## **Submitted by:**

Student Name : Gaus Saraf Murady

Student ID : 0432410005101088

Semester : Autumn, 2024

Batch : 55

Section : 2B2

Department : CSE

## **Tasks:**

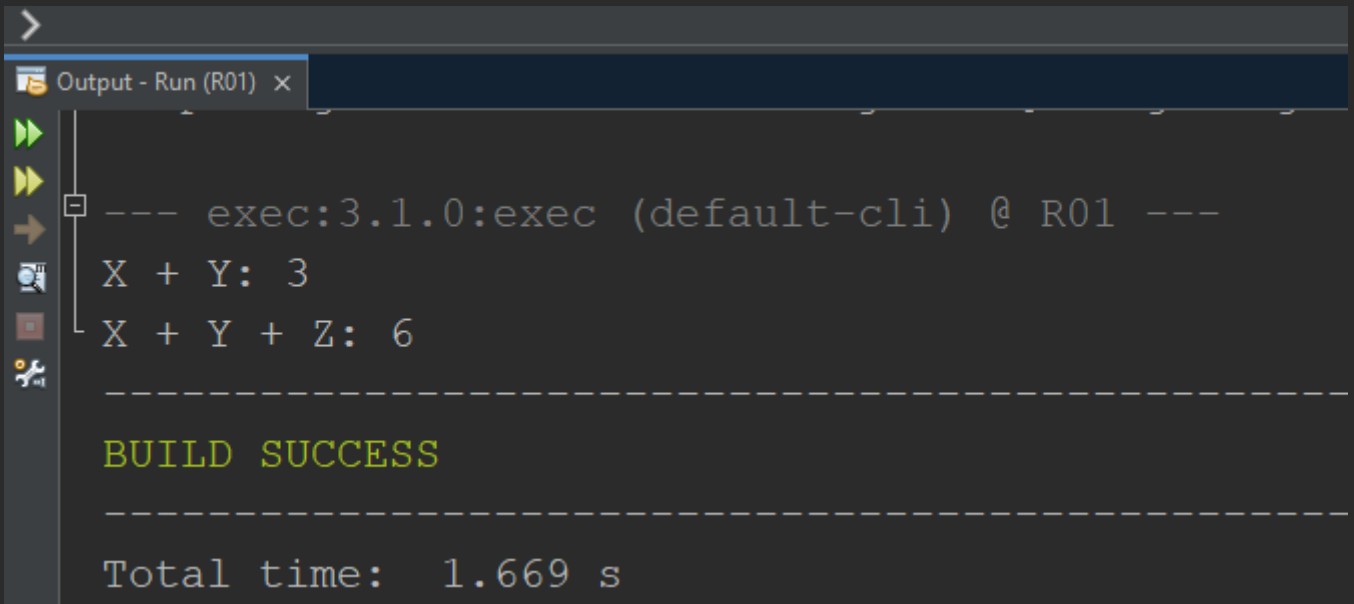
1. Implement single level inheritance
2. Implement multi level inheritance
3. Implement hybrid inheritance
4. Implement hierarchical inheritance

## Task 01: Implement single level inheritance

### Code:

```
R01.java x
Source History
1 package package01;
2
3 class A{
4     int x;
5     int y;
6     void displayXY(int x,int y){
7         System.out.println("X + Y: "+(x+y));
8     }
9 }
10
11 //Implementing single level inheritance
12
13 class B extends A{
14     int z;
15     void displayXYZ(int x,int y,int z){
16         System.out.println("X + Y + Z: "+(x+y+z));
17     }
18 }
19
20 public class R01 {
21     public static void main(String[] args) {
22         A object1 = new A();
23         B object2 = new B();
24         object1.displayXY(1,2);
25         object2.displayXYZ(1,2,3);
26     }
27 }
28 }
29
```

## Output:



```
>  
Output - Run (R01) x  
--- exec:3.1.0:exec (default-cli) @ R01 ---  
X + Y: 3  
X + Y + Z: 6  
-----  
BUILD SUCCESS  
-----  
Total time: 1.669 s
```

## Conclusion:

In this code, we have implemented single level inheritance using Java. We created two classes A and B and class B is the child class of parent class A. Therefore, objects under class B inherits the class variables of class A and prints the 2<sup>nd</sup> line of the output when the code is run.

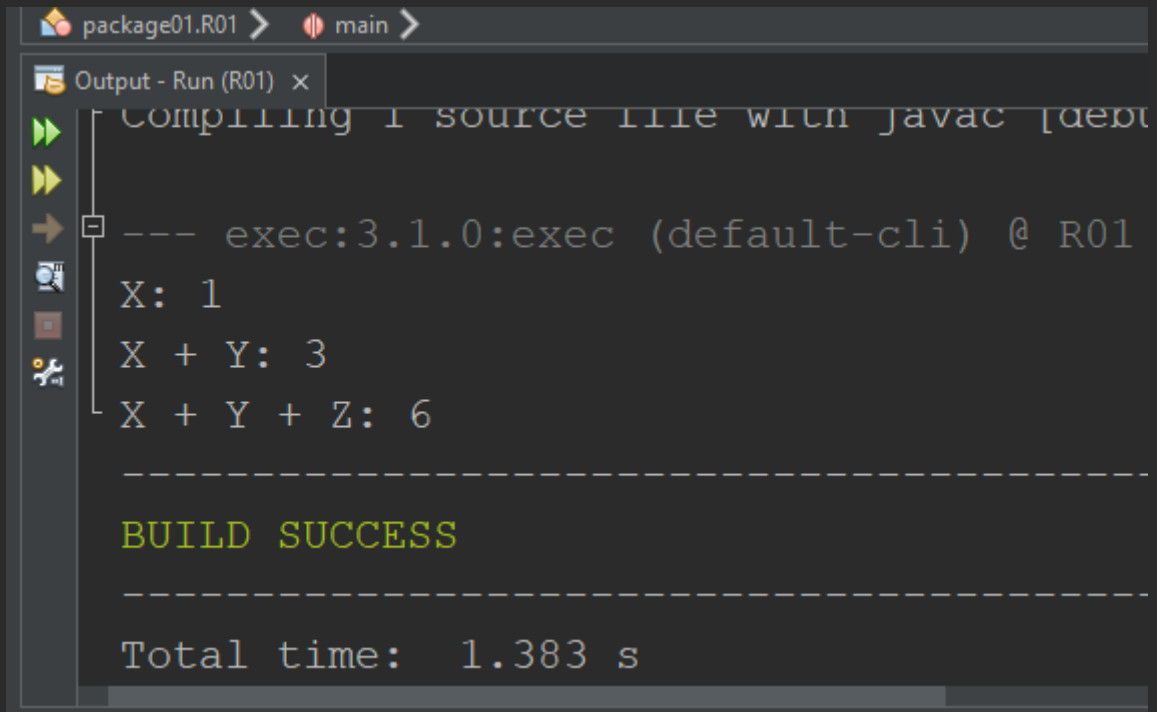
## Task 02: Implement multi level inheritance

Code:

```
R01.java x
Source History
1 package package01;
2
3 class A{
4     int x;
5
6     void displayX(int x){
7         System.out.println("X: "+x);
8     }
9 }
10
11 class B extends A{
12     int y;
13     void displayXY(int x,int y){
14         System.out.println("X + Y: "+(x+y));
15     }
16 }
17
18 //Implementing multi level inheritance
19
20 class C extends B{
21     int z;
22     void displayXYZ(int x,int y,int z){
23         System.out.println("X + Y + Z: "+(x+y+z));
24     }
25 }
26
```

```
24
25 public class R01 {
26     public static void main(String[] args) {
27         A object1 = new A();
28         B object2 = new B();
29         C object3 = new C();
30         object1.displayX(1);
31         object2.displayXY(1,2);
32         object3.displayXYZ(1,2,3);
33     }
34 }
35
```

## Output:



```
package01.R01 > main >
Output - Run (R01) x
Compiling 1 source file with javac [debu
--- exec:3.1.0:exec (default-cli) @ R01
X: 1
X + Y: 3
X + Y + Z: 6
BUILD SUCCESS
Total time: 1.383 s
```

## Conclusion:

In this code, we have implemented multi level inheritance using Java. We created a total of 3 classes A, B and C. Class C is the child class of parent class B, and class B is the child class of parent class A. Therefore, objects under class C inherits the class variables of class B and objects under class B inherits the class variables of class A. So, it prints the 2<sup>nd</sup> and 3<sup>rd</sup> line of the output when the code is run.

## Task 03: Implement hybrid inheritance

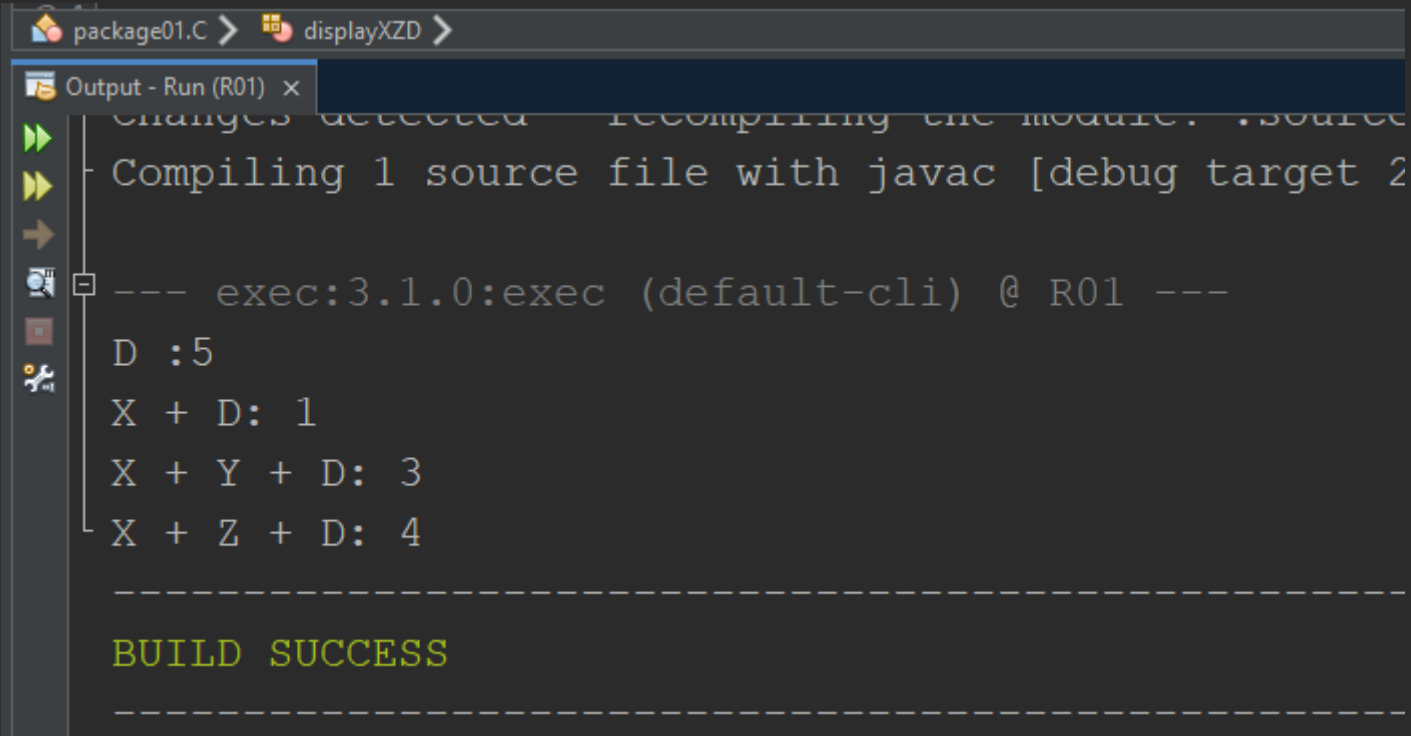
Code:

```
R01.java x
Source History
1 package package01;
2
3 class D{
4     int d;
5     void displayD(int d){
6         System.out.println("D :"+d);
7     }
8 }
9
10 class A extends D{
11     int x;
12     void displayXD(int x){
13         System.out.println("X + D: "+(x+d));
14     }
15 }
16
17 //Implementing hybrid inheritance: Multi-level
18
19 class B extends A{
20     int y;
21     void displayXYD(int x,int y){
22         System.out.println("X + Y + D: "+(x+y+d));
23     }
24 }
25
```





## Output:



The screenshot shows an IDE's output window for a Java project. The top bar indicates the current file is 'package01.C' and the package is 'displayXZD'. The output window title is 'Output - Run (R01)'. The output text shows a compilation step: 'Compiling 1 source file with javac [debug target 2...'. Below this, a separator line is followed by the execution command: '--- exec:3.1.0:exec (default-cli) @ R01 ---'. The execution output consists of four lines: 'D :5', 'X + D: 1', 'X + Y + D: 3', and 'X + Z + D: 4'. A dashed line separates this from the final status 'BUILD SUCCESS', which is also followed by a dashed line.

```
package01.C > displayXZD >
Output - Run (R01) x
Compiling 1 source file with javac [debug target 2...
--- exec:3.1.0:exec (default-cli) @ R01 ---
D :5
X + D: 1
X + Y + D: 3
X + Z + D: 4
-----
BUILD SUCCESS
-----
```

## Conclusion:

In this code, we have implemented hybrid inheritance using Java. We created a total of 4 classes A, B, C and D. Class B and C both are the child classes of parent class A. And class A is a child class of class D. Here class A extends class D and class B and class C extends class A. Therefore, applying both multi-level and hierarchical inheritance concepts in one package. Finally, two types of inheritances makes for the hybrid inheritance.

## Task 04: Implement hierarchical inheritance

Code:

R01.java x

Source History

```
1  package package01;
2
3  class A{
4      int x;
5
6      void displayX(int x){
7          System.out.println("X: "+x);
8      }
9  }
10
11 //Implementing hierarchical inheritance
12
13 class B extends A{
14     int y;
15     void displayXY(int x,int y){
16         System.out.println("X + Y: "+(x+y));
17     }
18 }
19
20 class C extends A{
21     int z;
22     void displayXYZ(int x,int z){
23         System.out.println("X + Z: "+(x+z));
24     }
25 }
26
```



## Output:

```
package01.C >
Output - Run (R01) x
--- exec:3.1.0:exec (default-cli) @ R01 ---
X: 1
X + Y: 3
X + Z: 4
-----
BUILD SUCCESS
-----
Total time: 1.365 s
Finished at: 2024-11-07T01:43:34+06:00
```

## Conclusion:

In this code, we have implemented hierarchical inheritance using Java. We created a total of 3 classes A, B and C. Class B and C both are the child classes of parent class A. Therefore, objects under both class B and class C inherits the class variables of class A. But class B and class C do not share any properties between each other. So, it prints the 2<sup>nd</sup> and 3<sup>rd</sup> line of the output when the code is run, where y and z variables are class variables of class B and class C respectively.