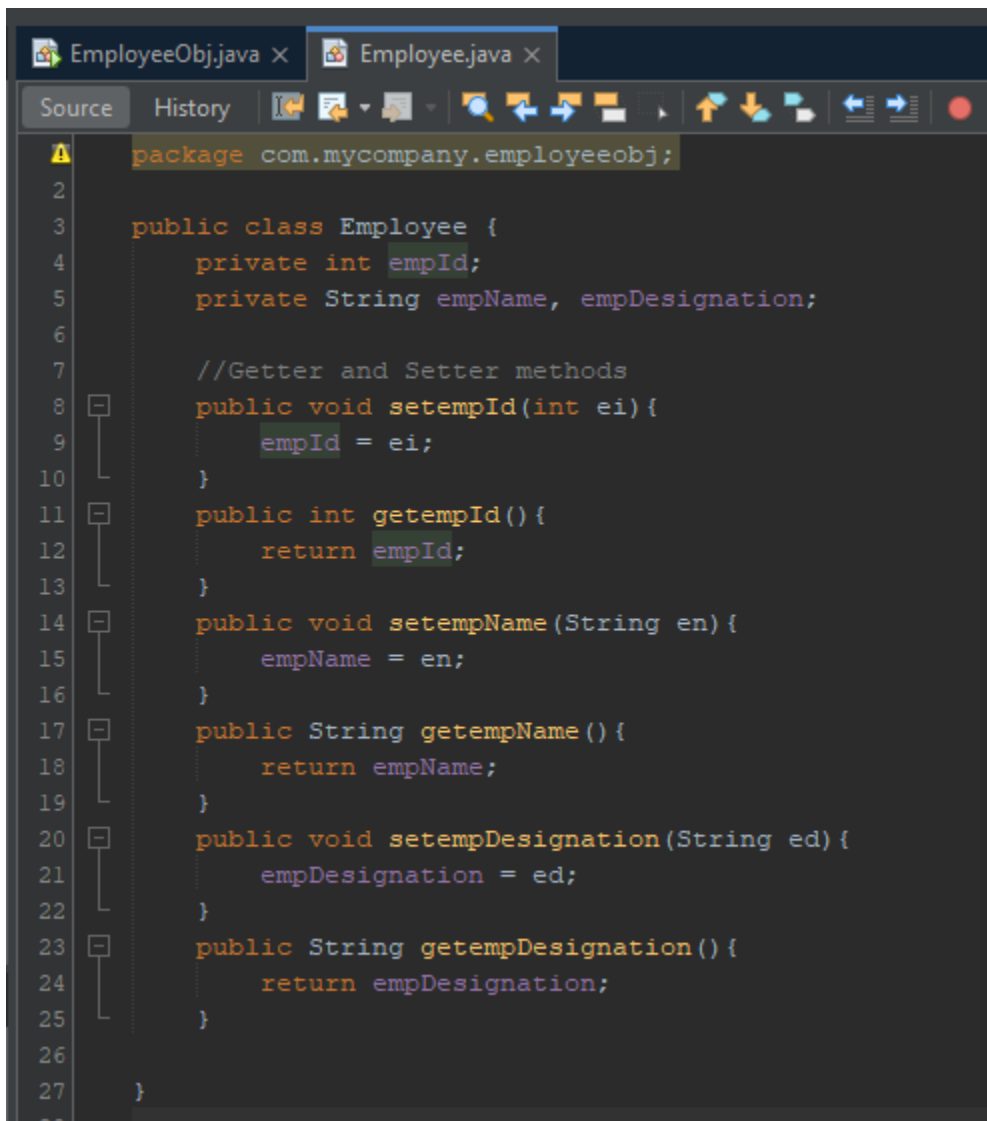


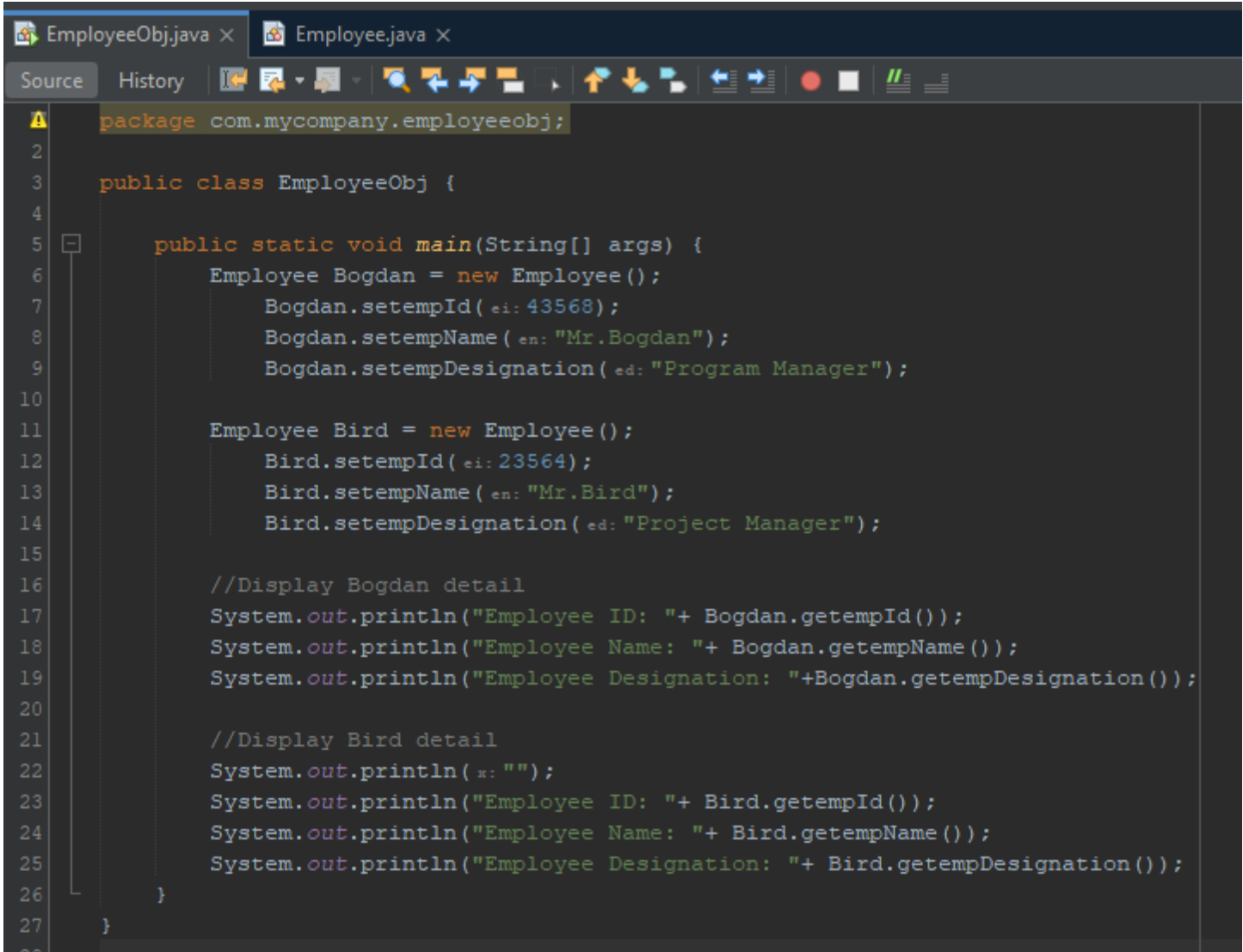
Practical 4.

Exercise 1;



```
EmployeeObj.java × Employee.java ×
Source History
package com.mycompany.employeeobj;

2
3 public class Employee {
4     private int empId;
5     private String empName, empDesignation;
6
7     //Getter and Setter methods
8     public void setempId(int ei){
9         empId = ei;
10    }
11    public int getempId(){
12        return empId;
13    }
14    public void setempName(String en){
15        empName = en;
16    }
17    public String getempName(){
18        return empName;
19    }
20    public void setempDesignation(String ed){
21        empDesignation = ed;
22    }
23    public String getempDesignation(){
24        return empDesignation;
25    }
26
27 }
```

A screenshot of an IDE window with two tabs: 'EmployeeObj.java' and 'Employee.java'. The 'EmployeeObj.java' tab is active, showing a Java source file. The code defines a package 'com.mycompany.employeeobj' and a public class 'EmployeeObj'. Inside the class, there is a 'main' method. This method creates two 'Employee' objects: 'Bogdan' and 'Bird'. 'Bogdan' is initialized with ID 43568, name 'Mr.Bogdan', and designation 'Program Manager'. 'Bird' is initialized with ID 23564, name 'Mr.Bird', and designation 'Project Manager'. After creating the objects, the code prints their details using 'System.out.println'. The output shows the employee ID, name, and designation for both Bogdan and Bird.

```
package com.mycompany.employeeobj;

public class EmployeeObj {

    public static void main(String[] args) {

        Employee Bogdan = new Employee();
        Bogdan.setempId( ei: 43568);
        Bogdan.setempName( en: "Mr.Bogdan");
        Bogdan.setempDesignation( ed: "Program Manager");

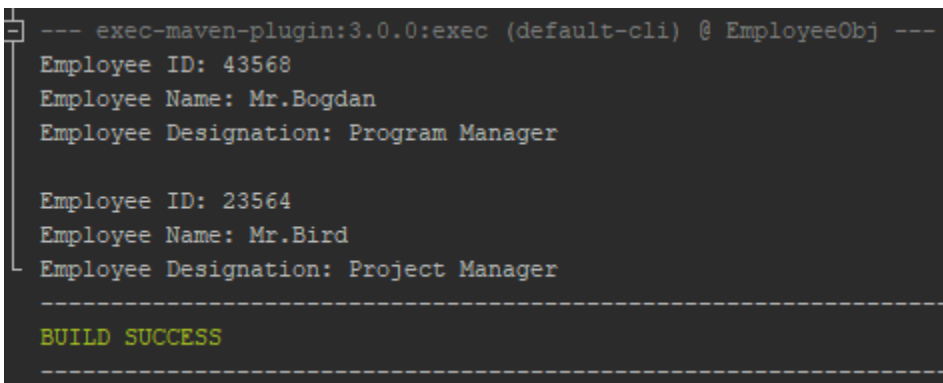
        Employee Bird = new Employee();
        Bird.setempId( ei: 23564);
        Bird.setempName( en: "Mr.Bird");
        Bird.setempDesignation( ed: "Project Manager");

        //Display Bogdan detail
        System.out.println("Employee ID: "+ Bogdan.getempId());
        System.out.println("Employee Name: "+ Bogdan.getempName());
        System.out.println("Employee Designation: "+Bogdan.getempDesignation());

        //Display Bird detail
        System.out.println( x: "");
        System.out.println("Employee ID: "+ Bird.getempId());
        System.out.println("Employee Name: "+ Bird.getempName());
        System.out.println("Employee Designation: "+ Bird.getempDesignation());

    }

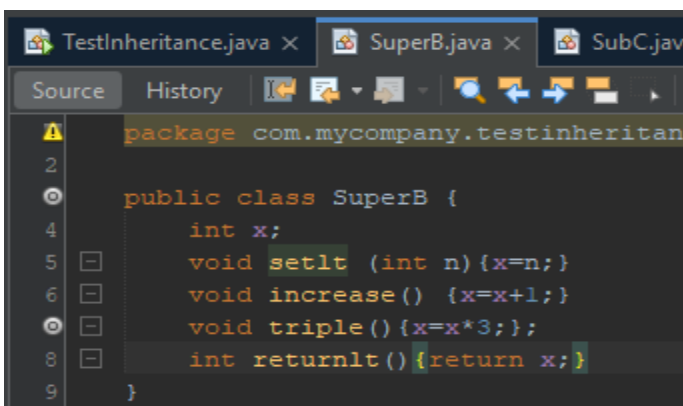
}
```

A screenshot of a terminal window showing the execution output of the 'EmployeeObj' class. The output is preceded by a separator line '--- exec-maven-plugin:3.0.0:exec (default-cli) @ EmployeeObj ---'. It displays the details for 'Bogdan' and 'Bird' as defined in the code, with each detail on a new line. The output ends with a separator line '-----' and the text 'BUILD SUCCESS' in green, followed by another separator line '-----'.

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ EmployeeObj ---
Employee ID: 43568
Employee Name: Mr.Bogdan
Employee Designation: Program Manager

Employee ID: 23564
Employee Name: Mr.Bird
Employee Designation: Project Manager
-----
BUILD SUCCESS
-----
```

Exercise 2;

A screenshot of an IDE window with three tabs: 'TestInheritance.java', 'SuperB.java', and 'SubC.java'. The 'SuperB.java' tab is active, showing a Java source file. The code defines a package 'com.mycompany.testinheritance' and a public class 'SuperB'. Inside the class, there is an integer variable 'x'. There are four methods: 'setlt' which takes an integer 'n' and sets 'x' to 'n'; 'increase' which increments 'x' by 1; 'triple' which multiplies 'x' by 3; and 'returnlt' which returns the value of 'x'.

```
package com.mycompany.testinheritance;

public class SuperB {

    int x;

    void setlt (int n){x=n;}
    void increase() {x=x+1;}
    void triple() {x=x*3;}
    int returnlt() {return x;}

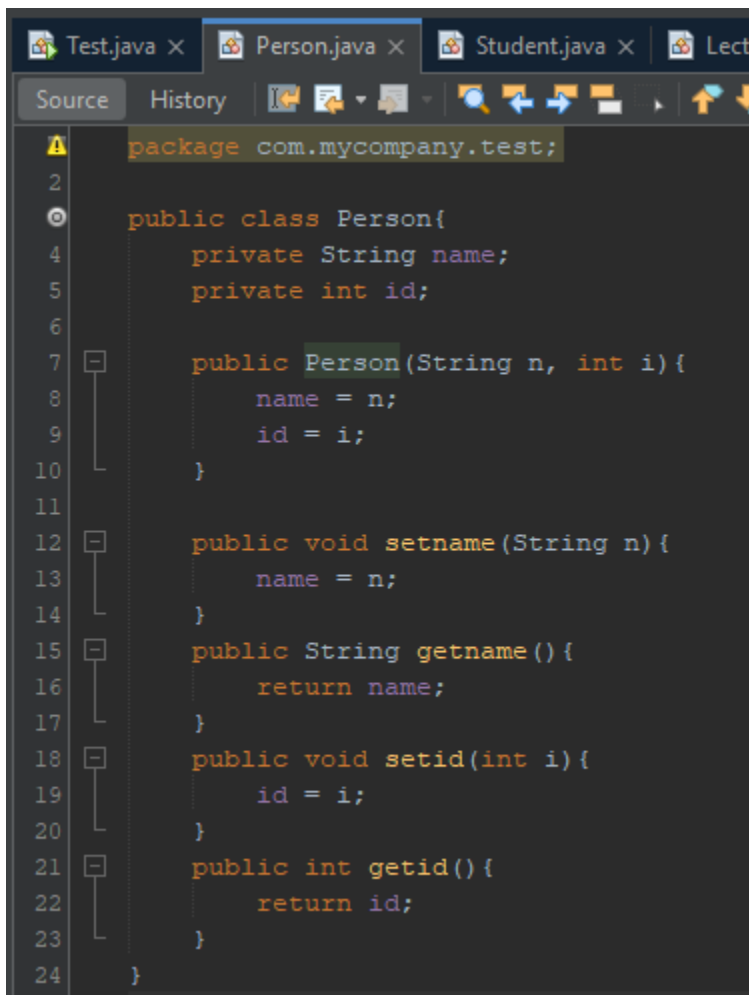
}
```

```
TestInheritance.java × SuperB.java × SubC.java ×
Source History
package com.mycompany.testinheritance;
2
3 public class SubC extends SuperB {
4     void triple() {x=x+3;} //override existing method
5     void quadruple() {x=x*4;} // new method
6 }
7
```

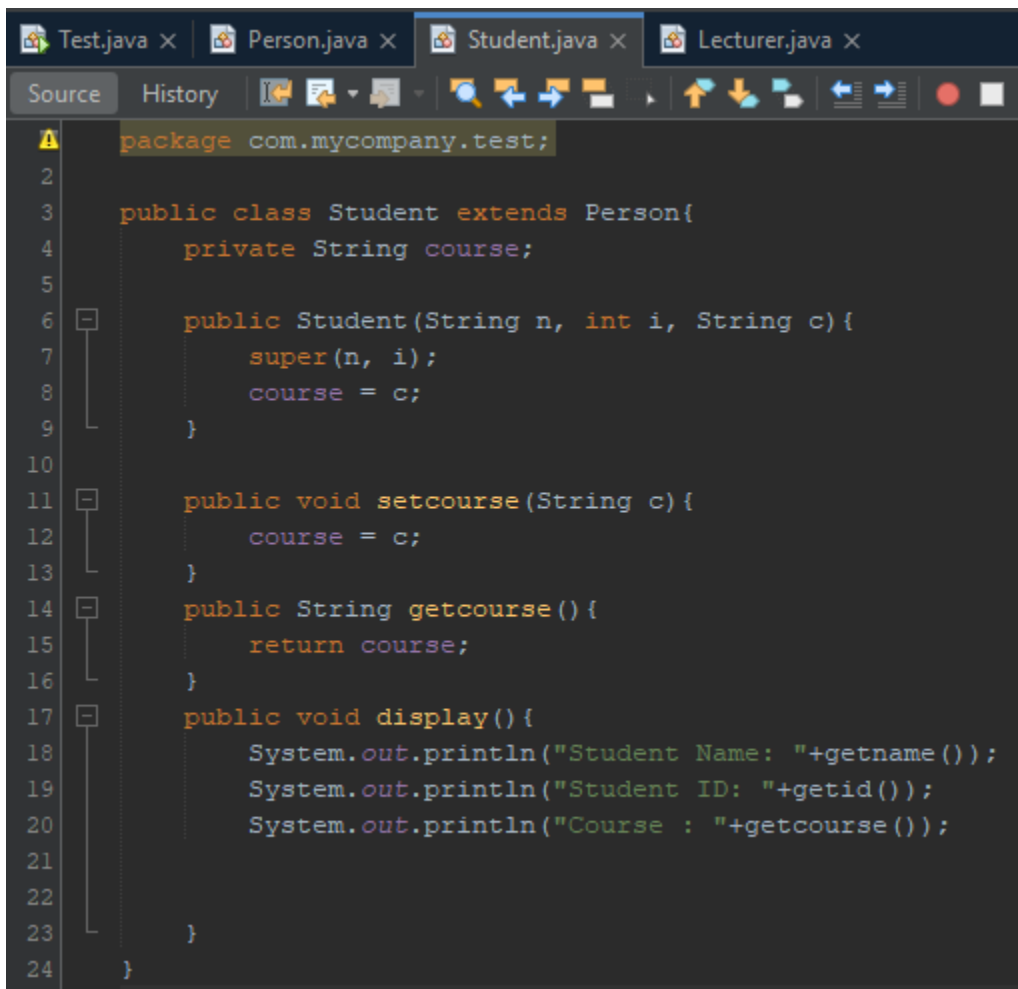
```
TestInheritance.java × SuperB.java × SubC.java ×
Source History
package com.mycompany.testinheritance;
2
3 public class TestInheritance {
4
5     public static void main(String[] args) {
6         SuperB b=new SuperB();
7         b.setIt(2);
8         b.increase();
9         b.triple();
10
11         System.out.println("b: "+b.returnIt());
12
13         SubC c=new SubC();
14         c.setIt(2);
15         c.increase();
16         c.triple();
17         System.out.println("c: "+c.returnIt());
18     }
19 }
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ TestInheritance ---
9
6
-----
BUILD SUCCESS
-----
```

Exercise 3;



```
Test.java x Person.java x Student.java x Lect
Source History
package com.mycompany.test;
2
3
4 public class Person{
5     private String name;
6     private int id;
7
8     public Person(String n, int i){
9         name = n;
10        id = i;
11    }
12
13    public void setname(String n){
14        name = n;
15    }
16
17    public String getname(){
18        return name;
19    }
20
21    public void setid(int i){
22        id = i;
23    }
24
25    public int getid(){
26        return id;
27    }
28 }
```



```
Test.java x Person.java x Student.java x Lecturer.java x
Source History
package com.mycompany.test;
2
3 public class Student extends Person{
4     private String course;
5
6     public Student(String n, int i, String c){
7         super(n, i);
8         course = c;
9     }
10
11    public void setcourse(String c){
12        course = c;
13    }
14
15    public String getcourse(){
16        return course;
17    }
18
19    public void display(){
20        System.out.println("Student Name: "+getname());
21        System.out.println("Student ID: "+getid());
22        System.out.println("Course : "+getcourse());
23    }
24 }
```

```
Test.java × Person.java × Student.java × Lecturer.java ×
Source History
package com.mycompany.test;

2
3 public class Lecturer extends Person{
4     private String programme;
5
6     public Lecturer(String n, int i, String p){
7         super(n,i);
8         programme = p;
9     }
10    public void setprogramme(String p){
11        programme = p;
12    }
13    public String getprogramme(){
14        return programme;
15    }
16    public void display(){
17        System.out.println("Lecturer Name: Mr./Mrs. "+getname());
18        System.out.println("Lecturer ID: "+getid());
19        System.out.println("Programme : "+getprogramme());
20    }
21 }
```

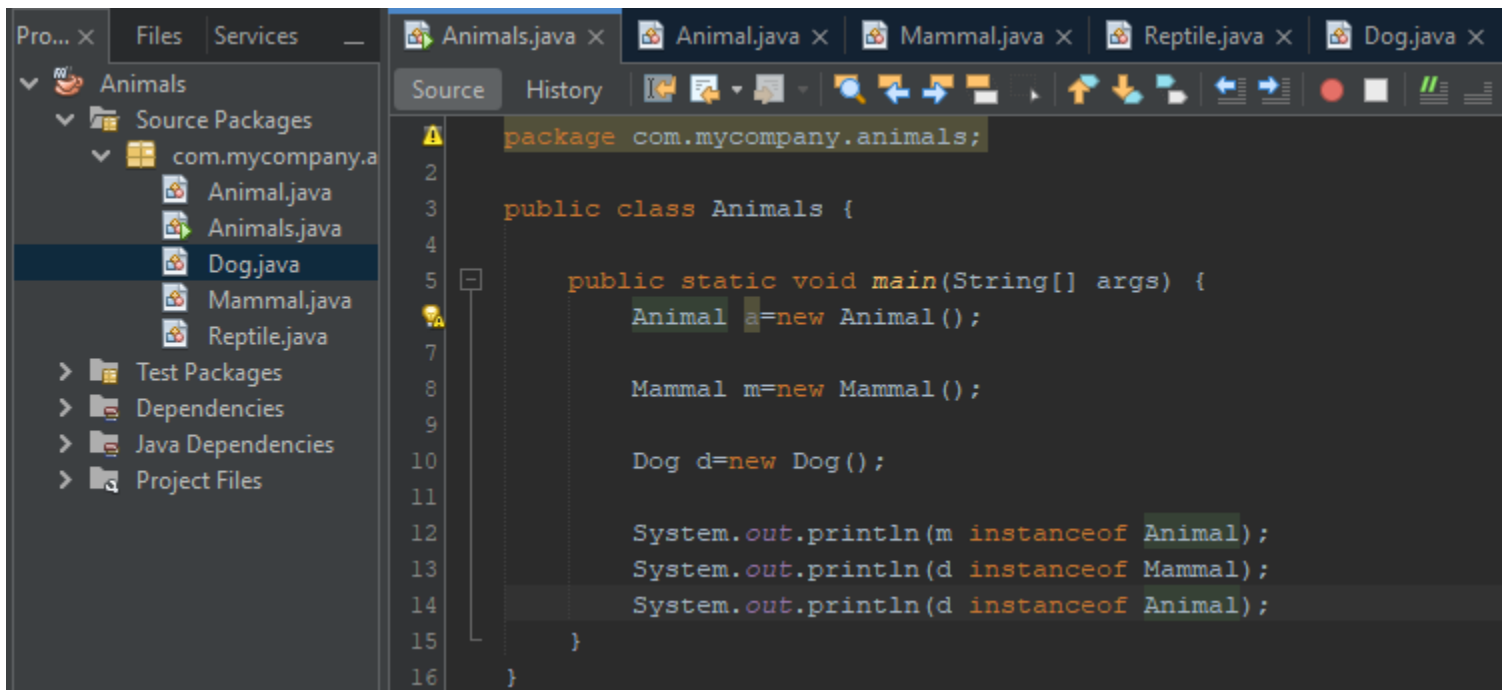
```
Test.java × Person.java × Student.java × Lecturer.java ×
Source History
package com.mycompany.test;

2
3 public class Test {
4
5     public static void main(String[] args) {
6         Student st=new Student( n: "Chathuranga", i:29214, c:"MIS");
7         Lecturer lr=new Lecturer( n: "P.D.Silva", i:45685, p: "Programming");
8
9         st.display();
10        System.out.println( x: "");
11        lr.display();
12    }
13 }
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Test ---
Student Name: Chathuranga
Student ID: 29214
Course : MIS

Lecturer Name: Mr./Mrs. P.D.Silva
Lecturer ID: 45685
Programme : Programming
-----
BUILD SUCCESS
```

Exercise 4;



```
package com.mycompany.animals;

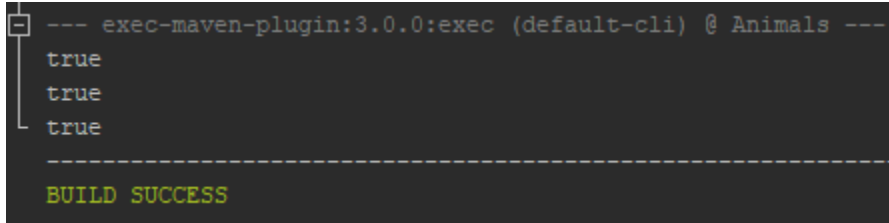
public class Animals {

    public static void main(String[] args) {
        Animal a=new Animal();

        Mammal m=new Mammal();

        Dog d=new Dog();

        System.out.println(m instanceof Animal);
        System.out.println(d instanceof Mammal);
        System.out.println(d instanceof Animal);
    }
}
```



```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Animals ---
true
true
true
-----
BUILD SUCCESS
```