

---

## **Lab 06**

### **Inheritance**

#### **Objective:**

The objective of this lab is to familiarize the students with various concepts and terminologies of inheritance using Java.

#### **Activity Outcomes:**

This lab teaches you the following topics:

- Declaration of the derived classes along with the way to access of base class members.
- Protected Access modifier and working with derived class constructors.

#### **Instructor Note:**

As pre-lab activity, read Chapter 11 from the text book “Introduction to Java Programming”, Y. Daniel Liang, Pearson, 2019.

---

## 1) Useful Concepts

### a. Inheritance

Inheritance is a way of creating a new class by starting with an existing class and adding new members. The new class can replace or extend the functionality of the existing class. The existing class is called the base class and the new class is called the derived class.

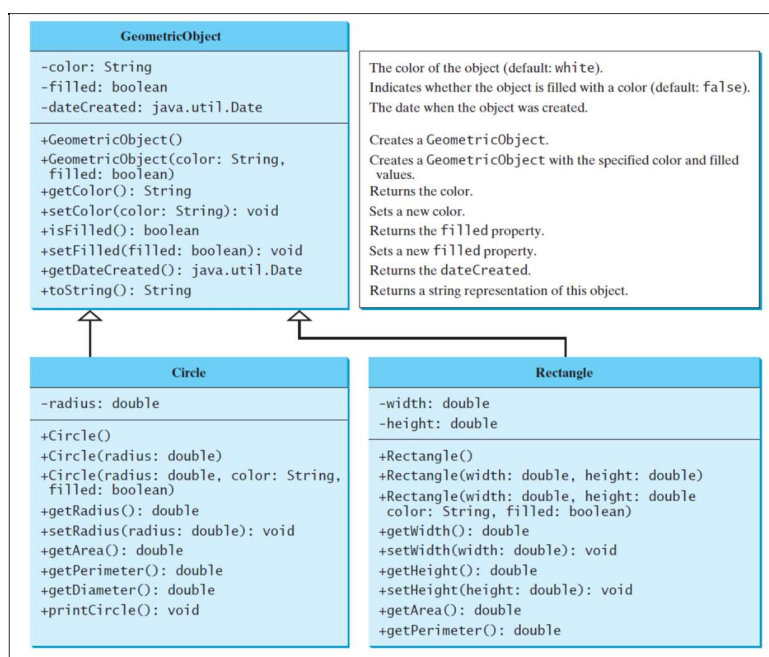
### b. Protected Access Specifier

Protected members are directly accessible by derived classes but not by other users.

A class member labeled **protected** is accessible to member functions of derived classes as well as to member functions of the same class.

### c. Derived class constructor

Constructors are not inherited, even though they have public visibility. However, the super reference can be used within the child's constructor to call the parent's constructor. In that case, the call to parent's constructor must be the first statement.



---

## 2) Solved Lab Activities (Allocated Time 50 min.)

<i>Sr.No</i>	<i>Allocated Time</i>	<i>Level of Complexity</i>	<i>CLO Mapping</i>
<i>Activity 1</i>	<i>25 mins</i>	<i>Medium</i>	<i>CLO-4</i>
<i>Activity 2</i>	<i>25 mins</i>	<i>Medium</i>	<i>CLO-4</i>

### Activity 1:

*This example will explain the method to specify the derived class. It explains the syntax for writing the constructor of derived class.*

### Solution:

```
public class person {  
  
    protected String name ; protected String id ; protected int phone ;  
  
    public person() {  
        name = "NaginaNazar" ; id = "sp14bcs039" ; phone = 12345 ;  
    }  
  
    public person(String a , String b , int c)  
    { name = a ; id = b ; phone = c ;}  
  
        public void setName(String a){ name = a ;}  
  
    public void setId(String j){id = j ;}  
  
    public void setPhone(int a) { phone = a ;}  
  
    public String getName() {return name ;}  
  
    public String getid() {return id ;}  
  
    public int getPhone() {return phone ;}
```

---

```
public void display( )
{
System.out.println("Name : " + name + "ID : " + id + "Phone : " +
phone ) ;}}
```

```
-----
public class student extends person {
private String rollNo;
private int marks ;

public student() {
super() ;
rollNo = "sp14bcs039" ; marks = 345 ;
}
public student(String a , String b , int c , String d , int e)
{ super(a,b,c) ;
rollNo = d ; marks = e ;
}
```

```
public void setRollNo(String a){ rollNo = a ;}
```

```
public void setMarks(int a ){ marks = a ;}
```

```
public String getRollNo() { return rollNo ;}
```

```
public int getMarks() {return marks ;}
```

```
public void display( )
{
super.display();
System.out.println("Roll # : " + rollNo + "\nMarks : " + marks) ;
}
}
```

```
-----
public class Runner
{
public static void main(String []args)
```

---

```
{
student s = new student ("Ahmed", "s-09", 123, "sp16-bcs-98",50);
s.display();
} }
```

## Output

Name : Ahmed

ID : s-09

Phone : 123

Roll # : sp16-bcs-98

Marks : 50

## Activity 2:

*This example demonstrates another scenario of inheritance. The super class can be extended by more than one class.*

```
public class Employee {

protected String name;
protected String phone;
protected String address;
protected int allowance;

public Employee(String name, String phone, String address, int
allowance)
{
this.name = name; this.phone = phone; this.address = address;
this.allowance = allowance;
}
}

-----

public class Regular extends Employee
{
private int basicPay;
```

---

---

```

public Regular(String name, String phone, String address, int
allowance, int basicPay)
{
    super(name, phone, address, allowance);
    this.basicPay = basicPay;
}

public void Display(){
    System.out.println("Name: " + name + "Phone Number: " + phone
+"Address: " + address + "Allowance: " + allowance + "Basic Pay: "
+ basicPay);
}
}

-----

public class Adhoc extends Employee
{
    private int numberOfWorkingDays; private int wage;

    public Adhoc(String name, String phone, String address,
int allowance, int numberOfWorkingDays, int wage)
    {
        super(name, phone, address, allowance);
        this.numberOfWorkingDays = numberOfWorkingDays;
        this.wage = wage;
    }

    public void Display()
    {
        System.out.println("Name: " + name + "Phone Number: " + phone
+"Address: " + address + "Allowance: " + allowance +
"Number Of Working Days: " + numberOfWorkingDays + "Wage: " +
wage);
    }
}

-----

public class Runner

```

---

```
{
public static void main(String []args){
Regular regularObj = new
Regular("Ahmed","090078601","Islamabad",15000,60000);
regularObj.Display();
Adhoc adhocObj = new
Adhoc("Ali","03333333333","Rawalpindi",500,23,1500);
adhocObj.Display();
}
}
```

## Output

Name: Ahmed Phone Number: 090078601 Address: Islamabad  
Allowance: 15000 Basic Pay: 60000

Name: Ali Phone Number: 03333333333 Address: Rawalpindi  
Allowance: 500 Number Of Working Days: 23 Wage: 1500

## 3) Graded Lab Tasks ( Allocated Time 2 Hr 10 Min.)

**Note: The instructor can design graded lab activities according to the level of difficulty and complexity of the solved lab activities. The lab tasks assigned by the instructor should be evaluated in the same lab.**

### Lab Task 1

*(The Person, Student, Employee, Faculty, and Staff classes)*

*Design a class named Person and its two subclasses named Student and Employee. Design two more classes; Faculty and Staff and extend them from Employee. The detail of classes is as under:*

*A person has a name, address, phone number, and email address.*

*A student has a status (String)*

*An employee has an office, salary, and date hired. Use the Date class to create an object for date hired.*

*A faculty member has office hours and a rank.*

*A staff member has a title.*

*Create display method in each class*

---

## Lab Task 2

*Imagine a publishing company that markets both book and audio-cassette versions of its works. Create a class publication that stores the title and price of a publication. From this class derive two classes:*

- i. book, which adds a page count and*
- ii. tape, which adds a playing time in minutes.*

*Each of these three classes should have set() and get() functions and a display() function to display its data. Write a main() program to test the book and tape class by creating instances of them, asking the user to fill in their data and then displaying the data with display().*

## Lab Task 3

*Write a base class Computer that contains data members of wordsize(in bits), memorysize (in megabytes), storagesize (in megabytes) and speed (in megahertz). Derive a Laptop class that is a kind of computer but also specifies the object's length, width, height, and weight. Member functions for both classes should include a default constructor, a constructor to initialize all components and a function to display data members.r.*