Task 1:

Make a class **Address** which has two private attribute city and country and one fully parametrized constructor.

Make a class **Person** that has two private attribute name and address. Attribute address should be of type Address. This class has fully parameterized constructor.

Make another class **Student**. Class Student is inherited from class Person. This class has three private attributes program, year and fee and one fully parametrized constructor.

Make a class **Staff** that is inherited from class Person. This class has one attribute that is pay and has one fully parameterized constructor.

In main function, create one object for Student and one for Staff. Ask user to input all information of Student and Staff and print values of all attributes.

All attributes are private.

Expected Output:

```
run:
Enter Student Information
Enter name:Ali
Enter city:Lahore
Enter country:Pakistan
Enter program:BSCS
Enter year:2019
Enter fee:85000
Enter Staff Information
Enter name:Maryam
Enter city:Islamabad
Enter country:Pakistan
Enter pay:80000

Student information is given below:
Name: Ali
Address: Lahore Pakistan
Program: BSCS
Year: 2019
Fee: 85000

Staff information is given below:
Name: Maryam
Address: Islamabad Pakistan
Pay: 80000
```

Task 2:

Make a class **Circle** that has two private attribute **color** and **radius**. Create a fully parameterized constructor. This class has a method *calculateArea()*, which calculates area of circle. This class is inherited by **Cylinder** that has one private attribute height. Create a fully parameterized constructor. This class overrides *calculateArea() method.*

In main function, call calculateArea() method of Cylinder. Put all classes in com.oop.lab.inheritance package.

Area of Cylinder= $2(\pi r^2 + \pi rh)$

Hint: πr^2= Area of Circle

Expected Output:

```
Enter radius:3.5
Enter height:2
Area of cylinder is 120.89
```

Task 3:

Make a class **Shape** that has one private attribute color. Create a fully paramterized constructor. This class has two public method *calculteArae()* and *calculatePerimeter().* Both methods return 0.

Make a class **Circle** that has one private attribute radius. Class Circle is inherited from class Shape. Create a fully parameterized constructor.

Make a class **Rectangle** that extends Shape, and it has two private attribute width and length. Create a fully parametrized constructor.

All sub classes overrides the *calculteArae()* and *calculatePerimeter().*

Declare the method of Shape class "calculateArea()" as final. Check the error. Declare the Shape class as final. Check the error.

In main function create ask user to input all values and display area and perimeter for Circle and Rectangle.

Expected Output:

```
run:
Enter color of circle:Black
Enter radius:3.5
Enter color of rectangle:Red
Enter length:5
Enter width:2

Color of Circle:Black
Area of Circle:38.465
Perimeter of Circle:21.98

Color of Rectangle:Red
Area of Rectangle:10.0
Perimeter of Rectangle:14.0
BUILD SUCCESSFUL (total time: 37 seconds)
```

Task 4:

Make a class **CommissionEmployee** that is inherited by **BasePlusCommissionEmployee.** Class **CommissionEmployee** has four private attribute firstName, LastName, grossSales and commissionRate. Create a fully parameterized constructor. In this class, you have to check that grossSales must be greater than zero and commissionRate must be greater than zero and less than 1. This class has 2 methods, *earnings()* and *toString().* You can calculate earnings of employee using this formula: commissionRate*grossSales.

Make a class **BasePlusCommissionEmployee** that has one private attribute baseSalary. Create a fully parameterized constructor. In this class, you have to check that baseSalary must be greater than 500. This class overrides both methods *earnings()* and *toString().* You can calculate earnings of employee using this formula: commissionRate*grossSales*baseSalary.

In the main function, get the firstName, lastName, grossSale, commissionRate and baseSalary from user and call *toString()* and *earnings()* methods.