Assignment No 1
(Classes and Objects)
Program:

```java
public class Complex{ float
    real,img;
    Complex(){          }
    Complex(float r,float i){
        real = r; img = i;
    }
    Complex        add(Complex        a){
        Complex temp = new Complex();
        temp.real  =  this.real  +  a.real;
        temp.img   =  this.img   +  a.img;
        return temp;
    }
    Complex        sub(Complex        a){
        Complex temp = new Complex();
        temp.real  =  this.real  -  a.real;
        temp.img   =  this.img   -  a.img;
        return temp;
    } public  static  void  main(String  args[]){
        Complex c1 = new Complex(5.6f,6.12f);
        Complex c2 = new Complex(12.6f,4.03f);
        Complex result;
        result        =        c1.add(c2);
        System.out.println("Addition    ...    "+result.real+"+"+result.img+"i");
        result        =        c1.sub(c2);
        System.out.println("Subtraction ... "+result.real+"+"+result.img+"i");
} }
```

Output:
java -cp /tmp/RGwcsL7qcy Complex
Addition .... +10.15i
Subtraction .... +2.0899997i

Assignment No 2
(Polymorphism)
Program:

```java
import java.util.Scanner;
class Publication{ String
title;   int   copies,price;
void saleCopy(){


    }
}


class Book{
    String title,author;
    int   copies,price;
    void  saleCopy(){
    price = 250;
        System.out.println("Total sale: "+copies*price);
    }
    void orderCopies(){
        Scanner  in  =  new  Scanner(System.in);
        System.out.println("Enter no. of copies");
        copies = in.nextInt();
    }
}


class Magazine{
    String title; int
    copies,price;
    void
    saleCopy(){

    }              void
    orderQty(){

    }
    void currentIssue(){

    }              void
    receiveIssue(){


    }
}


public class P{
    public static void main(String args[]){
        Book b = new Book();
        b.orderCopies();
```

```
        b.saleCopy();
    }
}
```
/*
Output:
(base) os@os-Vostro-3268:~$ javac P.java
(base) os@os-Vostro-3268:~$ java P
Enter no. of copies
5
Total sale: 1250
*/

(Inheritance)

Program:

```java
class Employee{ String
 name, add, mail; float
 id,  mobile,basic;  void
 salary()
{ float da,hra,pf,cf,gross;
 da = basic * 97/100;
 hra = basic * 10/100; pf
 = basic * 12/100; cf =
 basic * 0.1f/100;
  gross = basic + da + hra - pf - cf;
  System.out.println("Name: " + name);
  System.out.println("Basic Salary: " + basic);
  System.out.println("Gross Salary: " + gross); }
}


class Programmer extends Employee{
  Programmer(String name,int sal){
        this.name = name;  basic =
        sal;
  }
}
class Assistant_Professor extends Employee{
        Assistant_Professor(String name,int sal){
        this.name = name; basic = sal;
        }
}
class Associate_Professor extends Employee{
        Associate_Professor(String name,int
        sal){ this.name = name; basic = sal;
        }
}
class Professor extends Employee{
        Professor(String name,int
        sal){ this.name = name; basic
        = sal;
        }
}
public  class  Inheritance{  public  static  void
        main(String args[]){
        Assistant_Professor ast = new Assistant_Professor("Jai",40000); ast.salary();
        Associate_Professor aso = new Associate_Professor("Satyajeet",60000); aso.salary();
        Professor pro = new Professor("Ram",75000); pro.salary();
        Programmer pm = new Programmer("Akash",100000); pm.salary();
        }
}
/*
```

Output:
(base) os@os-Vostro-3268:~$ javac Inheritance.java
(base) os@os-Vostro-3268:~$ java Inheritance
Name: Jai
Basic Salary: 40000.0
Gross Salary: 77960.0
Name: Satyajeet
Basic Salary: 60000.0
Gross Salary: 116940.0
Name: Ram
Basic Salary: 75000.0
Gross Salary: 146175.0
Name: Akash
Basic Salary: 100000.0
Gross Salary: 194900.0
*/

(Dynamic Binding)

Program:

```java
import java.util.Scanner; abstract

class Shape{

        double     val1,val2;

void input() {

  Scanner    sc    =    new    Scanner(System.in);

System.out.println("Enter  first  value");     val1 =

sc.nextDouble();

  System.out.println("Enter  second  value");        val2 =

sc.nextDouble();

        }

        abstract void compute_area();

        }

        class Triangle extends Shape {

                void compute_area() {

                        double area;

                        area = 1.0f/2.0f * val1 * val2;

                        System.out.println("Triangle Area: " + area);

        }

} class Rectangle extends Shape

{

        void compute_area() {

                double area;

area = val1 * val2;

                System.out.println("Rectangle Area: " + area);

        }
```

```java
}
public class Dynamic {        public        static

void main(String args[]) {

        Shape s;

        Triangle t = new Triangle();

Rectangle r = new Rectangle();

        s = t;

        s.input();

        s.compute_area();

        s = r;

        s.input();

        s.compute_area();

    }

}
```

Output:

```
java -cp /tmp/WvFGZKJONF Dynamic
Enter first value
10
Enter second value
20
Triangle Area: 100.0
Enter first value 20
Enter second value
20
 Rectangle Area: 400.0
```

(Interface)

Program

```java
import java.io.*;
interface Vehicle { void
changeGear(int a); void
speedUp(int a); void
applyBrakes(int a);
}

class Bicycle implements Vehicle{
int speed;
int gear;
public void changeGear(int newGear){
gear = newGear;
}

public void speedUp(int increment){ speed
= speed + increment;
}

public void applyBrakes(int decrement){
speed = speed - decrement;
}

public void printStates() {
System.out.println("speed: " + speed
+ " gear: " + gear);
}
}

class Bike implements Vehicle {
int speed;
int gear;
public void changeGear(int newGear){
gear = newGear;
}

public void speedUp(int increment){ speed
= speed + increment;
}

public void applyBrakes(int decrement){
speed = speed - decrement;
```

```java
    }

    public void printStates() {
    System.out.println("speed: " + speed
    + " gear: " + gear);
    } } class
    StartVehicle {
    public static void main (String[] args) {
    Bicycle bicycle = new Bicycle();
    bicycle.changeGear(2);
    bicycle.speedUp(3);
    bicycle.applyBrakes(1);

    System.out.println("Bicycle present state :"); bicycle.printStates();

    Bike bike = new Bike();
    bike.changeGear(1);
    bike.speedUp(4);
    bike.applyBrakes(3);
    System.out.println("Bike present state :");
    bike.printStates();
    }
}
```

Output:

```
Bicycle present state:

speed: 2 gear: 2 Bike

present state: speed:

1 gear: 1
```