```
Package CIE;
import java.util.Scanner;
Public class internals {
  Public double cieMarks[];
  Public double cieSum = 0.0;
  int i;

Public void laccept () {
  Scanner in = new Scanner (System.in);
  cieMarks = new double [5];
  System.out.println (" Enter cie marks of 5
  courses out of 50 ");
  for( i=0; i<5; i++) {
    System.out.println ("Enter cie marks [" +(i+1)+ "]
      cieMarks [i] = in.nextDouble();
  }
}
}.
```

```
Package CIE;
import java.util.*;
Public class Student {
  Public String USN;
  Public String name;
  Public int sem;
```

```
Public void accept(){
Scanner in = new Scanner(System.in);
System.out.println("enter details usn");
usn = in.next();
name = in.next();
sem = in.nextInt();
}
Public void display(){
System.out.println("USN: "+usn"\n name: "+name+
"\n sem: "+sem);
}
}
import CIE.*;
import SEE.*;
import java.util.*;
Class FinalMarks{
Public static void main(String args[]){
int FS;
Scanner in = new Scanner(System.in);
System.out.println("Enter no. of students");
int n = in.nextInt();
CIE.Student[] cs = new CIE.Student[n];
CIE.Internals[] ci = new CIE.Student[n];
SEE.Externals[] se = new Externals[n];
```

```
for (int j=0; j<n; j++) {
cs[j] = new CIE.Student();
cs[j].accept();
ci[j] = new CIE.Internals();
ci[j].iaccept();
se[j] = new Externals();
se[j].Eaccept();
}
for(int j=0; j<n; j++){
cs[j].display();
System.out.println("Final marks of student
"+(j+1));
for(int i=0; i<5; i++){
FS = (int)(ci[j].CieMarks[i] + se[j].seeMarks[i]/2);
System.out.println("In course: "+(i+1)+"is: "+(FS));
}
}
}
```

```
Package SEE
import java.util.Scanner;
import CIE.*;
Public class Externals extends CIE.Student {
Public double seeMarks[];
int i;
```

```
Public void Eaccept() {.
  Scanner E = new Scanner (System.in);
  SeeMarks = new double[5];
  Credits = new int[5];
  System.out.println ("Enter See Marks of 5 courses
  out of 100");
    for (i=0; i<5; i++) {.
    System.out.println ("Enter See marks ["+(i+1)+"]");
    SeeMarks[i] = E.nextDouble();
    }
  }
}.
```

Program 7

```
class TwoGen<T> {.
  T obj1;
TwoGen (T o1) {.
  obj = o1;
}.
  void showtype() {.
  System.out.println ("Type of T is " + obj.getclass().
                                        getname());
  }
  T getobj() {.
  return obj;
}
```

```
class simple {
    Public static void main (String args[]) {
    TwoGen <Integer> tgobj = new TwoGen <Integer> (88);
    ;

    tgobj. showTypes ();
    int v = tgobj. getobj ();
    System. out. println ("value : " + v);
    TwoGen <String> tgObj2 = new TwoGen <String>
                              ("this is Prateek");

    tgobj2 . showtypes ();
    String str = tgobj2. get obj();
    System.out. println ("value : " + str);
    TwoGen <Float> ob3 = new TwoGen <Float> (23.23f);

     ob3. showtypes ();
     Float flo = ob3. getobj ();
     System. out. println ("value : " + flo);

  }
}
```

Program - 8 :

```
Class Father {
 static void acceptName F (int inputAge) throws
                                ArithmeticException.
 {
  try {
  if (inputAge<0){.
  throw new {ArithmeticException}.("Wrong age");
   }
 }
  catch (ArithmeticException e) {.
   System.out.println ("caught " +e);
 }!
};
Class son extends Father {
 static void checkSFage (int S_Age , int F_Age)
 throws ArithmeticException .{
 try {.
 if (S_Age >=F_Age)
 throw new ArithmeticException ("Son's age should
 be smaller than father's age");
 System.out.println (" Son's age is "+ S_.Age + " Father's
 age is " + F_Age);
}
 catch (ArithmeticException e) {.
   System.out.println(" Fcaught " +e);
}
```

```
}
}.
Public class Exception {.
  Public static void main (String args[]){.
    Father. acceptName(10);
    Son. checkSage(30, 20);
  }
}.
```

Program - 9.

```
class cse extends Thread {.
  cse() {.
    super("cse thread");
  System.out.println("The thread is:" + this);
    start();
}
Public void run() {
  try {
    for(int i=0; i<5; i++) {.
      System.out.println("CSE");
       Thread.sleep(2000);
  }
}
  catch (InterruptedException e) {.
    System.out.println("Thread interrupted");
  }
  System.out.println("cse thread exiting");
}
}.
```

```
class Main {
  Public static void main(String args[]) {
    new cse ();
    try {
      for(int i=0; i<5; i++) {
        System.out.println("BMS college of Engineering");
        Thread.sleep(2000);
      }
    }
    catch (InterruptedException e) {
      System.out.println("Main thread interrupted");
    }
    System.out.println("Main thread exiting");
  }
}
```

Program -10.

```
class car_queue {
    String n;
    boolean valueset = false;
  synchronized String get() {
      while (!valueset) {
        try {
          wait();
        }
        catch (InterruptedException e) { }
  }
```

```
System.out.println("Got: " + n);
valueSet = false;
notify();
return n;
}
synchronized void put(String n) {
while(valueSet) {
try {
wait
} catch(InterruptedException e) { }
this.n = n; valueSet = true;
System.out.println("Put: " + n);
notify();
}
}
class Car_Owner implements Runnable {
Car_queue q;
Car_Owner(Car_queue q) {
this.q = q;
new Thread(this, "Owner").start();
}
Public void run() {
while(true) {
q.Put("Brakes");
}
}
}
Class Car_Mechanic implements Runnable {
Car_queue q;
Car_Mechanic(Car_queue q)
this.q = q;
```

```
new Thread(this, "Mechanic").start();
}
Public void run() {
    while(true) {
    q.getit();
    }
}
}
class inter {
  Public static void main(String args[]){
    Car_queue q = new Car_queue();
    new Car_Owner(q);
    new Car_queue(q);
    System.out.println("Press control c to stop");
  }
}
```