

ASSIGNMENT #2

*WAQAS ASHIQ.*

*BCS-F11-201.*

*SECTION C.*

*SUBMITTED TO: MADAM AYESHA SADIQ.*

**/\*Implementation Of Polymorphism And Abstract Classes\*/**

**CODE:**

abstract class person{ **//Super Abstract Class**

String Name;

String NIC;

int Age;

person(String N,String Ni,int A){ **//Parameterized Constructor**

Name=N;

NIC=Ni;

Age=A;

}

}

abstract class Student extends person{  **//Extended Abstract Class**

String Regno;

String Prog;

int Sem;

Student(String N,String Ni,int A,String R,String P,int S){ **//Parameterized Constructor**

super(N,Ni,A);

Regno=R;

Prog=P;

Sem=S;

}

abstract double CalculateFee(); **//Abstract Methods**

abstract void DisplayStudentInfo();

}

class UnderGraduateStudent extends Student{ **//Extended Concrete Class**

UnderGraduateStudent(String N,String Ni,int A,String R,String P,int S){

super(N,Ni,A,R,P,S);

} **//Parameterized Constructor**

double CalculateFee(){ **//Concrete Methods**

if(Sem==1){

return 43000+20000;

}

else{

return 43000;

}

}

void DisplayStudentInfo(){

System.out.println("Student Name="+Name+", Reg# = "+Regno+", Program\_Enrolled= "+Prog+", Semester = "+Sem+", Fee = "+CalculateFee());

}

}

class GraduateStudent extends Student{ **//Extended Concrete Class**

GraduateStudent(String N,String Ni,int A,String R,String P,int S){

super(N,Ni,A,R,P,S);

} **//Parameterized Constructor**

double CalculateFee(){

if(Sem==1){

return 70000+16000; **//Concrete Methods**

}

else{

return 70000;

}

}

void DisplayStudentInfo(){

System.out.println("Student Name="+Name+", Reg# = "+Regno+", Program\_Enrolled= "+Prog+", Semester = "+Sem+", Fee = "+CalculateFee());

}

}

class StudentDummy{ **//Main Class**

public static void main (String[] args) {

UnderGraduateStudent Ug=new UnderGraduateStudent("Waqas Ashiq","35202",19,"BCS-F11-201","BSCS",3);

GraduateStudent G=new GraduateStudent("Noman Ali","35202",24,"MS-F12-123","MS",1);

**//Created Reference Of Concrete Classes**

Ug.DisplayStudentInfo();

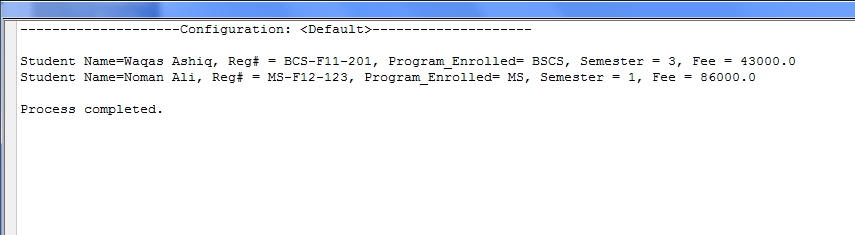
G.DisplayStudentInfo();

**//Methods Called**

}

}

**OUTPUT:**

****

**CLASS DIAGRAM:**

Super Abstract class person

Attributes:Name,NIC,Age

Methods: No local Method

Class Graduate Extends Student

Attribute:No Local Variable

Methods:Calculatefee,

DisplayInfo

Class Undergraduate Extends Student

Attributed: No Local Variable

Methods:Calculatefee,

DisplayInfo

Abstracted Class Studen Extends Person

Attributes:Reg#,Prog,Sem

Methods:Calculatefee,Displayinfo(Abstract)