

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No. | INDEX  Program | Page  No. | Teacher’s  signature  Date |  |
| 1. | Write a program to create a class ‘Student’ with the student details (roll no, name,marks,branch etc). Create two functions to input and ouput the data through objects of the class. | 1 | 04/02/2019 |  |
| 2. | Write the program to demonstrate the use of default constructor | 3 | 11/02/2019 |  |
| 3. | Write a program to demonstrate the use of parameterised constructor | 4 | 18/02/2019 |  |
| 4. | Write a program to demonstrate the use of copy constructor | 5 | 18/02/2019 |  |
| 5. | Employees have a number, dob, rank and salary. When an employee is first recruited then all these are given values of 0. Upon confirmation, the actual values of these are entered for the employee. Employees can be given a salary rise of 10% of their previous salary. Their rank can be incremented by 1 and when this happens an employee gets an increment of 25%. Write a C++ class for employee. Define constructors and destructors wherever required. | 6 | 25/02/2019 |  |
| 6. | Write a program that uses inheritance to create two subclasses(truck and automobile) of a given class(road\_vehicle). Implement using both multi-level and multiple inheritance. | 9 | 11/03/2019 |  |
| 7. | Using the concept of polymorphism write a program to calculate area of shapes (circles,rectangles,squares,triangle). | 12 | 18/03/2019 |  |
| 8. | Draw use case diagram of ATM Machine using C++ graphics. | 13 | 25/03/2019 |  |
| 9. | Draw use case diagram of railway reservation system using C++ graphics. | 17 | 25/03/2019 |  |
| 10. | Draw Collaboration diagram of 3 use cases of ATM Machine. | 20 | 01/04/2019 |  |
| 11. | Draw class diagram of university registration system using StarUML | 24 | 08/04/2019 |  |
| 12. | Draw sequence diagram of three use cases of Railway System | 25 | 15/04/2019 |  |

**PROGRAM 1**

**Q.** Write a program to create a class ‘Student’ with the student details (roll no, name,marks,branch etc). Create two functions to input and ouput the data through objects of the class.

**STUDENT CLASS WITH STUDENTS DETAILS**

**TWO FUNCTIONS TO INPUT AND OUTPUT THE DATA THROUGH OBJECTS OF THE CLASS**

#include<iostream>

using namespace std;

class Student

{

int roll; // data members

char name[20];

char branch[10];

int marks[5];

public:

getDetails(); // member functions

DisplayDetails();

};

Student::getDetails() // member function defined outside the class

{

cout<<"enter roll no.:\n";

cin>>roll;

cout<<"enter name :\n";

cin>>name;

cout<<"enter branch:\n";

cin>>branch;

cout<<"enter marks of 5 subjects:\n";

for(int i=0;i<5;i++)

{

cin>> marks[i] ;

}

}

Student::DisplayDetails() // member function defined outside the class

{

cout<<"\nroll no :" <<roll<<endl;

cout<<"name:"<<name<<endl;

cout<<"branch:"<<branch<<endl;

cout<<"marks:";

for(int i=0;i<5;i++)

{

cout<< " "<< marks[i] ;

}

}

int main() // execution starts here

{

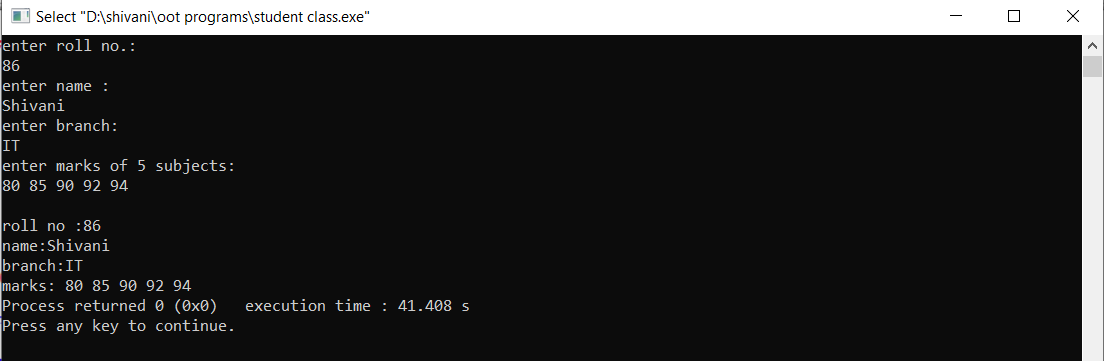
Student s; // object created

s.getDetails();

s.DisplayDetails();

return 0;

}



**PROGRAM 2**

**Q2** Write the program to demonstrate the use of default constructor

**DEMONSTRATION OF DEFAULT CONSTRUCTOR**

#include<iostream>

using namespace std;

class A

{

public:

int b,a;

A() // default constructor

{

a=2;

b=7;

}

};

int main()

{

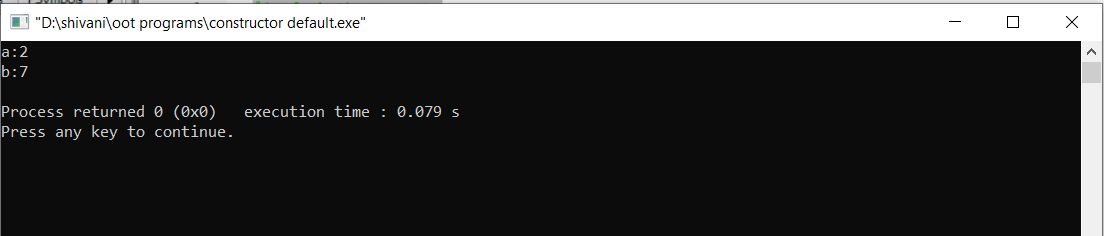
A g; // calls constructor

cout<<"a:"<<g.a<<endl;

cout<<"b:"<<g.b<<endl;

return 0;

}



**PROGRAM 3**

**Q2.** Write a program to demonstrate the use of parameterised constructor

**DEMONSTRATION OF PARAMETRISED CONSTRUCTOR WITH DESTRUCTOR**

#include<iostream>

using namespace std;

class A

{

int s,t;

public:

A(int a,int b) // parameterised constructor

{

s=a;

t=b;

cout<<"objects created \n";

}

void display() // member function

{

cout<<"s:"<<s<<endl;

cout<<"t:"<<t<<endl;

}

~A() // destructor

{

cout<<"Objects destroyed ";

}

};

int main() // execution starts here

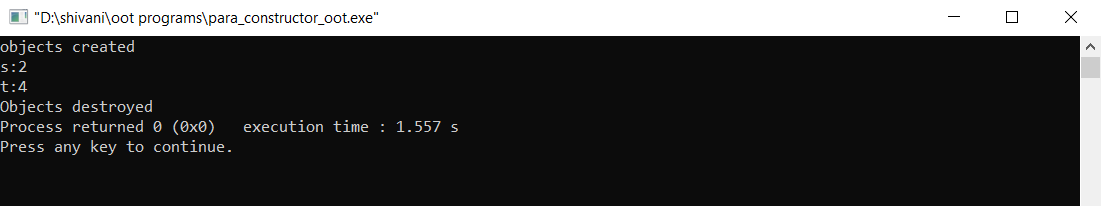
{

A l(2,4); // constructor called

l.display();

return 0;

}



**PROGRAM 4**

**Q** Write a program to demonstrate the use of copy constructor

**DEMONSTRATION OF COPY CONSTRUCTOR WITH DESTRUCTOR**

#include<iostream>

using namespace std;

class A

{

private:

int x, y; //data members

public:

A(int a, int b)

{

x = a;

y = b;

}

A(const A &p) // Copy constructor

{

x = p.x;

y = p.y;

}

void display()

{

cout<<x<<" "<<y<<endl;

}

};

int main() /\* main function \*/

{

A obj1(10, 15); // Normal constructor

A obj2 = obj1; // Copy constructor

cout<<"Normal constructor : ";

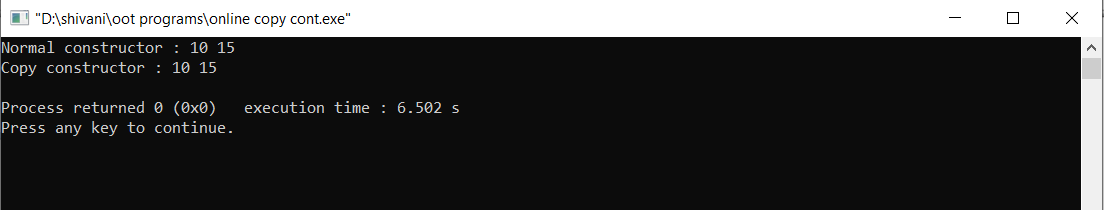
obj1.display();

cout<<"Copy constructor : ";

obj2.display();

return 0;

}



**PROGRAM-3**

**Q.** Employees have a number, dob, rank and salary. When an employee is first recruited then all these are given values of 0. Upon confirmation, the actual values of these are entered for the employee. Employees can be given a salary rise of 10% of their previous salary. Their rank can be incremented by 1 and when this happens an employee gets an increment of 25%. Write a C++ class for employee. Define constructors and destructors wherever required.

#include<iostream>

#include<conio.h>

using namespace std;

struct date

{

int dd;

int mm;

int yy;

};

class Employee

{

string name;

int number;

date dob;

int ran;

float salary;

public:

Employee()

{

name="xxxx";

number=0;

dob.dd=0;

dob.mm=0;

dob.yy=0;

ran=0;

salary=0;

}

void conformation()

{

cout<<"Enter name: ";

cin>>name;

cout<<"enter employee number: ";

cin>>number;

cout<<"enter employee rank: ";

cin>>ran;

cout<<"enter date of birth: ";

cin>>dob.dd>>dob.mm>>dob.yy;

cout<<"enter salary: ";

cin>>salary;

}

void increment(int in)

{

if(in==10)

{

salary=1.1\*salary;

}

if(in==25)

{

salary=1.25\*salary;

ran=ran+1;

}

}

void display()

{

cout<<name<<endl<<number<<endl<<ran<<endl<<salary<<endl<<dob.dd<<"/"<<dob.mm<<"/"<<dob.yy<<endl;

}

~Employee(){};

};

int main()

{

Employee \*e;

int n,i,k;

cout<<"Enter the number of employes: ";

cin>>n;

char b[n];

e= new Employee[n];

for(i=0; i<n; i++)

{

cout<<"if the job of employee is comfirmed enter 'Y' else enter 'N' ";

cin>>b[i];

if(b[i]=='Y'){

e[i].conformation();

cout<<"enter increment either 10% or 25%: ";

cin>>k;

e[i].increment(k);

}

}

for(i=0;i<n;i++){

if(b[i]=='Y')

e[i].display();

else{

e[i].~Employee();

cout<<"not recruited"<<endl;}

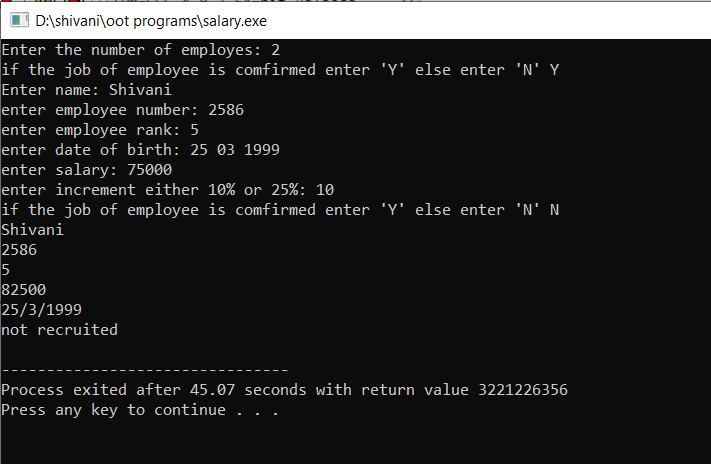
}

delete e;

getch();

return 0;

}



**PROGRAM 5**

**Q.** Write a program that uses inheritance to create two subclasses(truck and automobile) of a given class(road\_vehicle). Implement using both multi-level and multiple inheritance.

**MULTI LEVEL AND MULTIPLE INHERITANCE**

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<iostream>

using namespace std;

class road\_vehicle{

protected:

int wheels;

public:

int regno;

void setvalue(){

cout<<"Enter registration number ";

cin>>regno;

cout<<"Enter number of wheels ";

cin>>wheels;

}

void print1(){

cout<<"\nRegistration number= "<<regno;

cout<<"\nNumber of wheels= "<<wheels;

}

};

class automobile{

public:

int cc;

string color;

int seats;

void getvalue()

{

int x,y;

string c;

cout<<"Enter displacement of the engine ";

cin>>cc;

cout<<"Enter colour of the vehicle ";

cin>>color;

cout<<"Enter the number of seats ";

cin>>seats;

}

void print2(){

cout<<"\nDisplacement of the engine= "<<cc;

cout<<"\nColour of the vehicle= "<<color;

cout<<"\nNumber of seats= "<<seats;

}

};

class truck:public automobile,public road\_vehicle{

protected:

public:

int load;

int mileage;

void values(){

setvalue();

getvalue();

cout<<"Enter the capacity of the truck ";

cin>>load;

cout<<"Enter the total distance it has been driven(km) ";

cin>>mileage;

}

void printall(){

print1();

print2();

cout<<"\nTruck load="<<load<<"\nMileage"<<mileage;

}

};

int main(){

int size;

cout<<"\nEnter the number of entries: ";

cin>>size;

truck \*t= new truck[size];

for(int i=0;i<size;i++){

t[i].values();

}

int no;

int flag=0;

char check='y';

while(check=='y'||check=='Y'){

cout<<"\nEnter the registration number to be searched: ";

cin>>no;

for(int i=0;i<size;i++){

if(t[i].regno==no)

t[i].printall();

flag=1;

}

if (flag==0)

{cout<<"\nnot found";

}

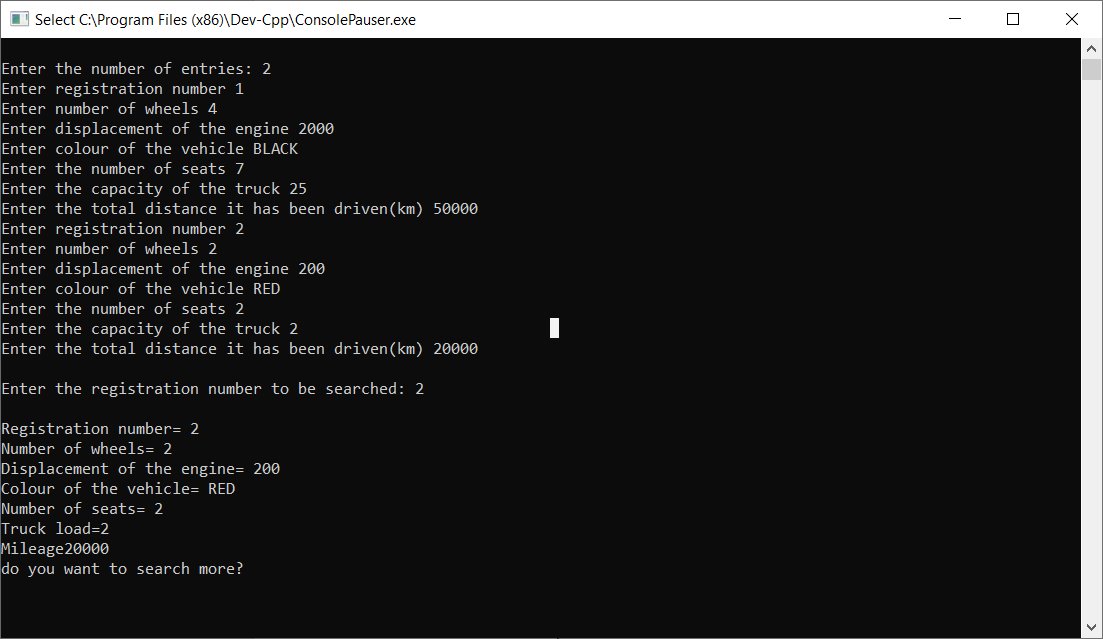
cout<<"\ndo you want to search more? ";

cin>>check;

}

return 0;

}



**PROGRAM 6**

**Q.** Using the concept of polymorphism write a program to calculate area of shapes (circles,rectangles,squares,triangle).

**AREA OF DIFFERENT SHAPES USING POLYMORPHISM**

#include<iostream>

#include<conio.h>

#include<cmath>

using namespace std;

class shape

{

int s;

int w;

public:

void area(int l){

cout<<l\*l<<endl;

}

void area(double r){

cout<<3.14\*r\*r<<endl;

}

void area(int l,int b){

cout<<l\*b<<endl;

}

void area(int a,int b,int c){

s=(a+b+c)/2;

w=s\*(s-a)\*(s-b)\*(s-c);

cout<<pow(w,0.5)<<endl;

}

};

int main()

{

shape tri,sq,ci,rec;

tri.area(3,3,4);

sq.area(7);

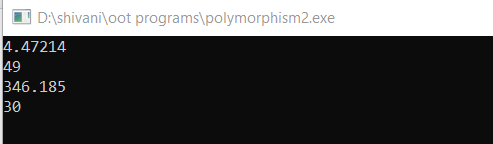
ci.area(10.5);

rec.area(5,6);

getch();

return 0;

}



**PROGRAM 7(A)**

**Q.** Draw use case diagram of ATM Machine using C++ graphics.

**USE CASE DIAGRAM OF ATM MACHINE**

#include<graphics.h>

#include<iostream.h>

#include<dos.h>

#include<conio.h>

void main()

{

int gdriver=DETECT,gmode;

initgraph(&gdriver,&gmode,"c:\\turboc3\\bgi");

outtextxy(280,40,"ATM SYSTEM");

circle(20,100,10);

line(20,110,20,150);

line(20,150,10,160);

line(20,150,30,160);

line(10,120,20,130);

line(30,120,20,130);

rectangle(100,10,550,450);

ellipse(170,100,0,360,60,30);

outtextxy(130,100,"insert card");

ellipse(280,170,0,360,60,30);

outtextxy(260,170,"deposit");

ellipse(280,240,0,360,60,30);

outtextxy(260,240,"withdraw");

ellipse(280,310,0,360,60,30);

outtextxy(260,310,"transfer");

line(40,120,110,100);

line(40,130,220,170);

line(40,140,220,240);

line(40,150,220,310);

line(230,100,260,140);

outtextxy(250,100,"include");

ellipse(440,200,0,360,60,30);

outtextxy(400,200,"print slip");

outtextxy(10,170,"USER");

outtextxy(340,160,"extend");

line(340,180,380,190);

circle(620,100,10);

line(620,110,620,150);

line(620,120,660,140);

line(600,130,620,120);

line(620,150,660,170);

line(600,160,620,150);

outtextxy(600,170,"BANK");

line(340,240,390,220);

outtextxy(330,210,"extend");

line(340,300,420,230);

outtextxy(370,280,"extend");

ellipse(400,350,0,360,60,30);

outtextxy(360,350,"maintanence");

ellipse(300,390,0,360,60,30);

outtextxy(250,390,"check balance");

line(420,320,600,140);

line(500,200,600,120);

line(40,180,260,370);

line(340,300,355,300);

line(340,300,335,285);

line(340,240,355,250);

line(340,240,345,225);

line(340,180,350,190);

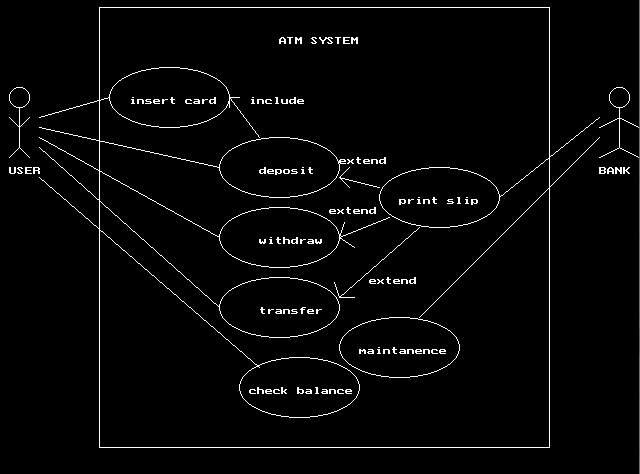
line(340,180,350,170);

line(230,100,240,100);

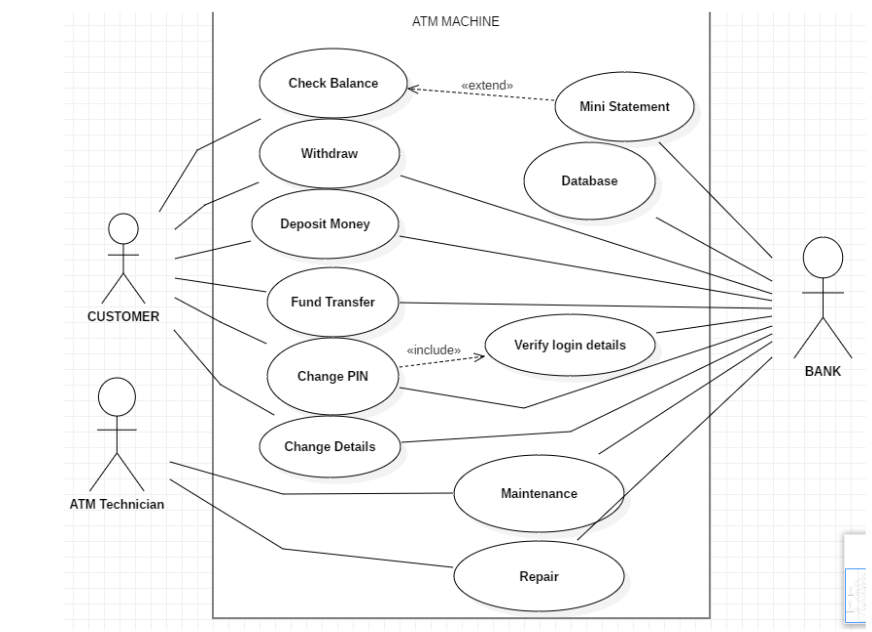
line(230,100,230,110);

getch();

}



**USE CASE DIAGRAM OF ATM USING STAR UML**



**PROGRAM 7(B)**

**Q.** Draw use case diagram of railway reservation system using C++ graphics.

**USE CASE DIAGRAM OF RAILWAY RESERVATION SYSTEM**

#include<conio.h>

#include<iostream.h>

#include<graphics.h>

void main()

{

int gmode,gdriver;

initgraph(&gdriver,&gmode,"c:\\turboc3\\bgi");

int maximumX=getmaxx();

int maximumY=getmaxy();

cout<<maximumX<<endl;

cout<<maximumY<<endl;

rectangle(80,40,540,450);

circle(10,80,10);

line(10,90,10,130);

line(10,100,20,110);

line(10,130,20,140);

line(0,140,10,130);

line(0,110,10,100);

outtextxy(200,50,"RAILWAY RESERVATION SYSTEM");

outtextxy(230,90,"enquire ticket");

outtextxy(230,100,"availability");

ellipse(280,100,0,360,70,30);

ellipse(450,150,0,360,50,20);

outtextxy(420,150,"fill form");

ellipse(180,160,0,360,60,30);

outtextxy(130,160,"book tickets");

ellipse(300,240,0,360,60,30);

outtextxy(250,240,"pay fare tax");

ellipse(200,300,0,360,60,30);

outtextxy(170,300,"print form");

ellipse(320,350,0,360,60,30);

outtextxy(270,350,"cancel ticket");

ellipse(450,400,0,360,60,30);

outtextxy(400,400,"refund money");

circle(10,350,10);

line(10,360,10,400);

line(10,360,20,380);

line(0,380,10,370);

line(0,410,10,400);

line(10,400,20,410);

circle(610,120,10);

line(610,130,610,170);

line(600,180,610,170);

line(610,170,620,180);

line(600,150,610,140);

line(610,140,620,150);

outtextxy(2,150,"USER");

outtextxy(580,200,"RAILWAY SYSTEM");

outtextxy(2,420,"CLERK");

line(25,80,210,90);

line(350,90,580,150);

line(500,150,580,150);

line(240,180,580,180);

line(360,230,580,180);

line(260,300,580,190);

line(380,340,580,200);

line(450,370,580,210);

line(30,370,140,310);

line(30,380,260,360);

line(25,90,120,160);

line(25,140,240,240);

line(25,140,180,270);

line(380,360,400,380);

outtextxy(390,350,"<<extend>>");

line(260,310,270,330);

outtextxy(270,310,"<<extend>>");

line(350,320,450,170);

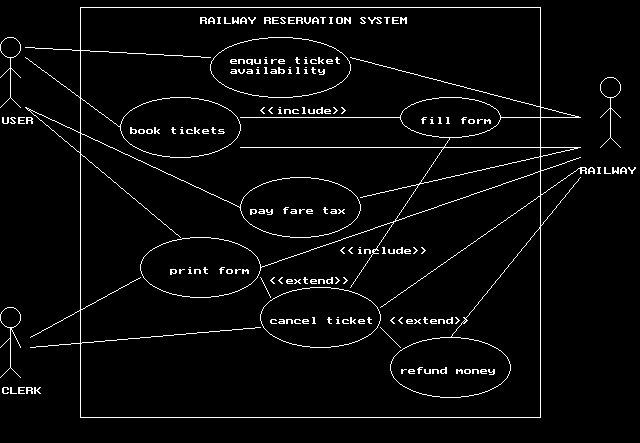
outtextxy(340,280,"<<include>>");

line(240,150,400,150);

outtextxy(260,140,"<<include>>");

getch();

}



**PROGRAM 10**

**Q.** Draw Collaboration diagram of 3 use cases of ATM Machine.

**COLLABORATION DIAGRAM FOR ANY 3 USE CASES OF ATM MACHINE**

#include<iostream.h>

#include<graphics.h>

#include<conio.h>

int main()

{

int gmode=DETECT,gdriver;

initgraph(&gmode,&gdriver,"c:\\turboc3\\bgi");

rectangle(20,100,100,130);

outtextxy(30,110,"Account");

rectangle(260,100,360,130);

outtextxy(270,110,"ATM Machine");

line(30,120,90,120);

line(270,120,360,120);

line(100,120,260,120);

outtextxy(120,80,"5:Process Transaction()");

outtextxy(120,140,"8:Transaction Successful()");

rectangle(520,100,620,130);

outtextxy(530,110,"Bank Client");

line(530,120,620,120);

line(360,120,530,120);

outtextxy(390,30,"2:Enter Kind()");

outtextxy(390,50,"4:Enter Amount()");

outtextxy(390,70,"13:Terminate()");

line(430,100,480,100);

line(420,100,430,90);

line(420,100,430,110);

line(430,90,430,110);

line(50,130,140,300);

rectangle(100,300,240,330);

outtextxy(110,310,"Checking Account");

line(110,320,240,320);

outtextxy(110,210,"Withdraw from checking account()");

outtextxy(0,280,"Withdrawal Successful()");

outtextxy(390,170,"1:Request Kind()");

outtextxy(390,190,"3:Request Amount()");

outtextxy(390,210,"9:Dispense cash()");

outtextxy(390,230,"10:Request take cash()");

outtextxy(390,250,"11:Take cash()");

outtextxy(390,270,"12:Request continuation()");

outtextxy(390,290,"14:Print Receipt()");

line(420,140,470,140);

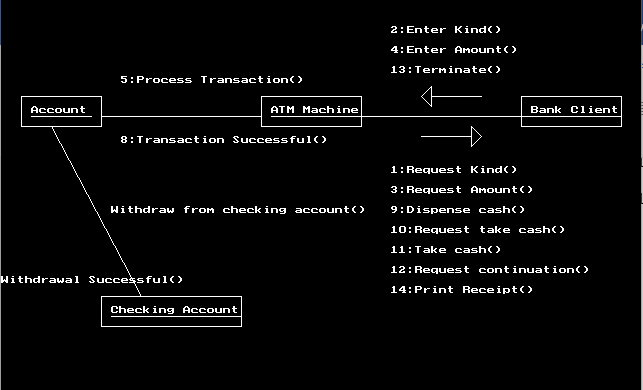
line(470,130,480,140);

line(470,150,480,140);

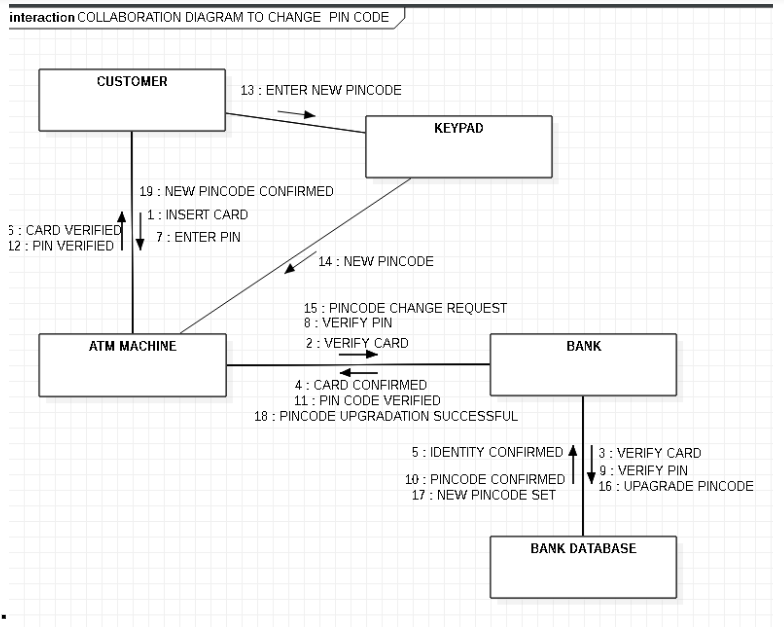
line(470,130,470,150);

getch();

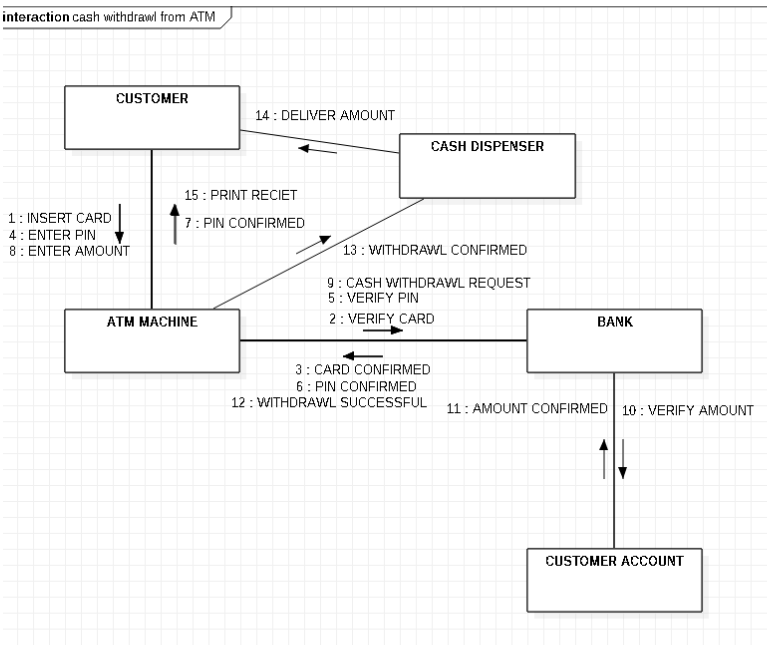
return 0;

}

**USING STARUML COLLABORATION DIAGRAM FOR second USE CASE**



**USING STARUML COLLABORATION DIAGRAM FOR THIRD USE CASE**



**PROGRAMS (done in lab) 25/03/2019**

**GRAPHICS.H**

#include<conio.h>

#include<graphics.h>

void main()

{

int gdriver=DETECT,gmode;

initgraph(&gdriver,&gmode,”c:\\turboc3\\bgi”);

rectangle(100,100,300,300);

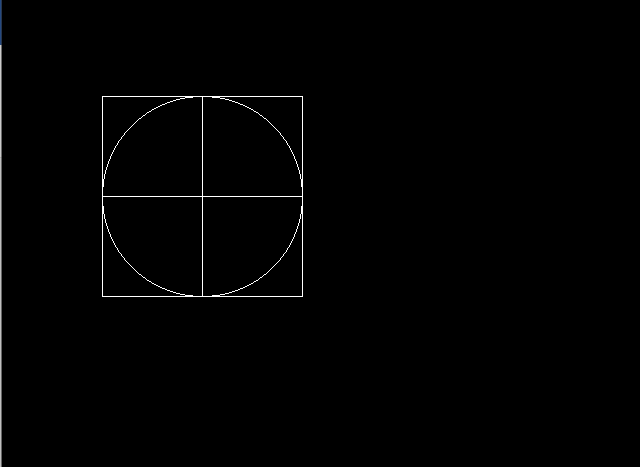
circle(200,200,100);

line(100,200,300,200);

line(200,100,200,300);

getch();

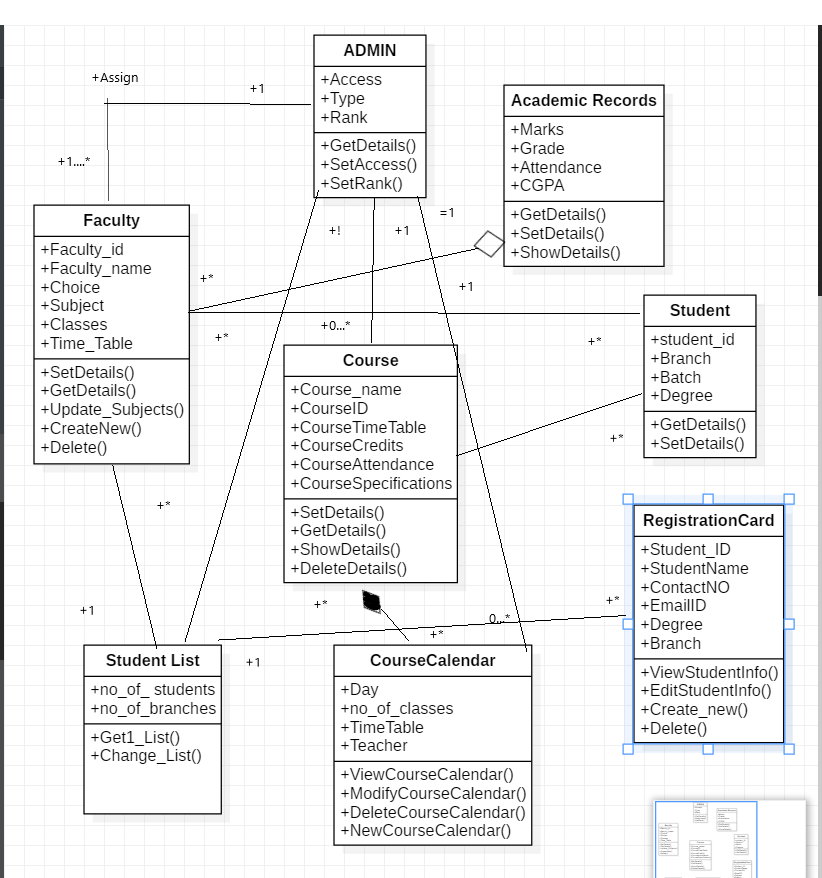
}



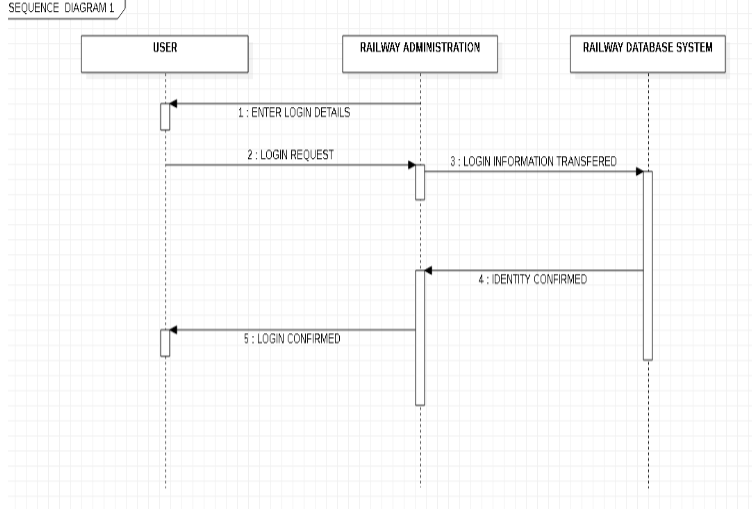
**Draw class diagram of university registration system using StarUML**

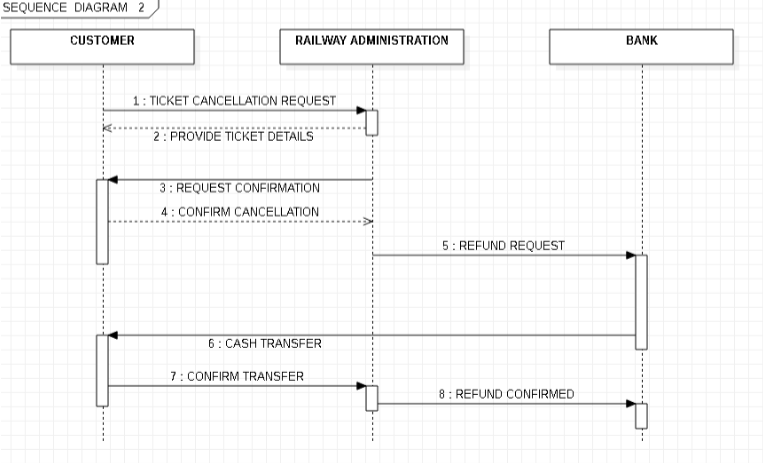
**CLASS DIAGRAM**

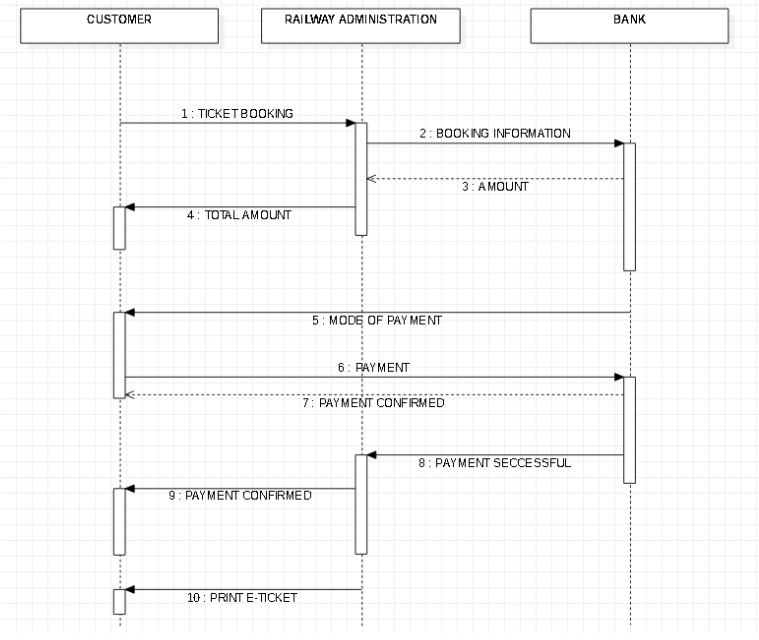
**UNIVERSITY REGISTRATION SYSTEM**



Draw sequence diagram of three use cases of Railway System

**1.**

**2.**

**3.**