# TEST

**AIM : 1 Write a C++ program to print factorial series using recursion.**

**Program :**

**#include<iostream>**

**using namespace std;**

**int fac(int n)**

**{**

**if(n==0 || n==1)**

**{**

**return 1;**

**}**

**else**

**{**

**return n\*fac(n-1);**

**}**

**}**

**int main()**

**{**

**int n;**

**cout << "====================================" << endl;**

**cout << "Enter the number: ";**

**cin >> n;**

**cout << "====================================" << endl << endl << endl;**

**cout << ".....Answer Of Factorial Number....." << endl << endl;**

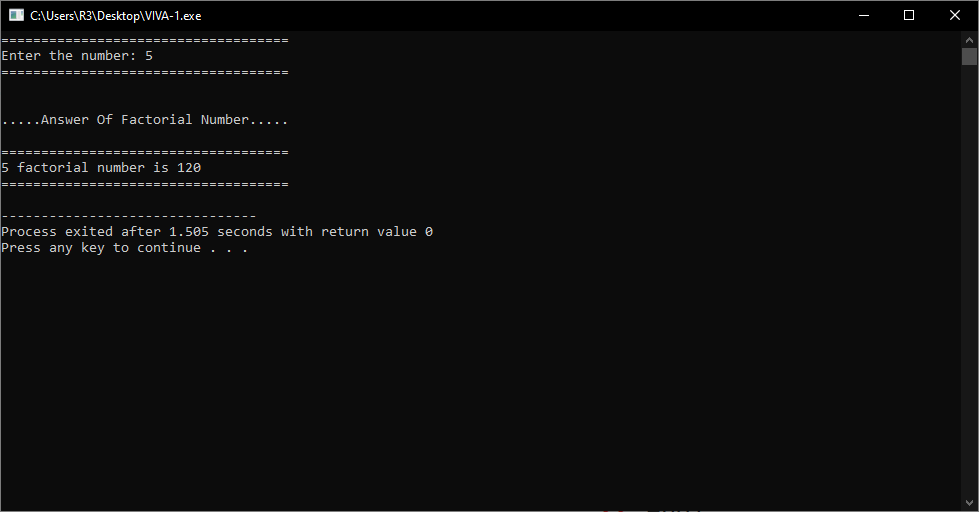
**cout << "====================================" << endl;**

**cout << n << " factorial number is " << fac(n) << endl;**

**cout << "====================================" << endl;**

**}**

**Output :**

****

**AIM :2 Array Of Object (Bank Management System) Requirements(A/C Number , A/C Name , A/C Type , A/C Branch , A/C Balance ) Search Record By A/C Number.**

**Program :**

**#include<iostream>**

**using namespace std;**

**class bank**

**{**

**public :**

**int Id;**

**string Name;**

**string Type;**

**string Branch;**

**int Balance;**

**public:**

**void setdata(int i, int n)**

**{**

**system("cls");**

**cout<<"Enter Bank Datil.."<<endl;**

**cout<<endl;**

**cout<<"Enter Id : ";**

**cin>>Id;**

**cout<<"Enter Name : ";**

**cin >> Name;**

**cout<<"Enter Type : ";**

**cin>>Type;**

**cout<<"Enter Branch name : ";**

**cin>>Branch;**

**cout<<"Enter Balance : ";**

**cin>>Balance;**

**}**

**void search(int s)**

**{**

**int i;**

**if(s==Id)**

**{**

**cout<<"Id : "<<Id<<endl;**

**cout<<"Name : "<<Name<<endl;**

**cout<<"Type : "<<Type<<endl;**

**cout<<"Branch name : "<<Branch<<endl;**

**cout<<"Balance : "<<Balance<<endl;**

**}**

**}**

**};**

**int main()**

**{**

**int i,n=1,s;**

**bank b[n];**

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

**{**

**b[i].setdata(i+1,n);**

**}**

**system("cls");**

**cout<<"Enter your Account Number : ";**

**cin >> s;**

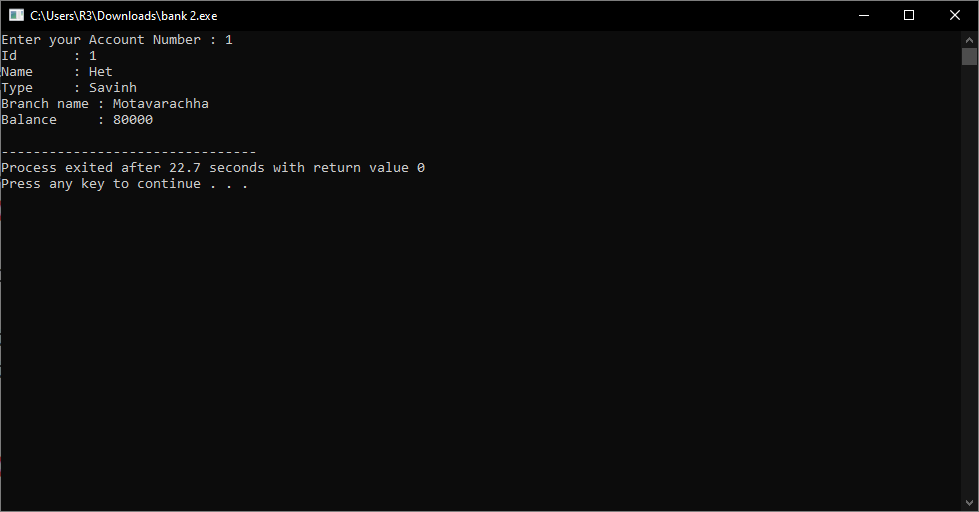
**for(i=0;i<n;i++){**

**b[i].search(s);**

**}**

**}**

**Output :**



**AIM : 3 Give an example which shows Exception handling of division by zero exception.**

**Program :**

**#include<iostream>**

**using namespace std;**

**class Division{**

**int a;**

**int b;**

**public:**

**void SetData()**

**{**

**cout << "Enter A : ";**

**cin >> a;**

**cout << "Enter B : ";**

**cin >> b;**

**}**

**void GetData()**

**{**

**if(b == 0)**

**{**

**throw b;**

**}**

**else**

**{**

**cout << "Division : " << a-b << endl;**

**}**

**}**

**void Div()**

**{**

**try**

**{**

**GetData();**

**}**

**catch(int b)**

**{**

**cout << b << " Can Not Divided By Zero" << endl;**

**}**

**}**

**};**

**int main()**

**{**

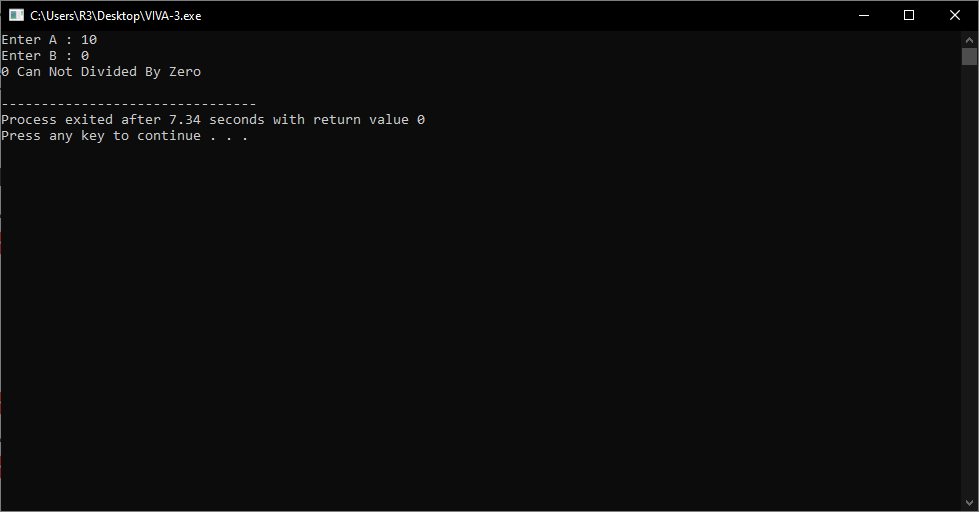
**Division d1;**

**d1.SetData();**

**d1.Div();**

**}**

**Output :**

****

**AIM : 4 Write a Program which illustrates Method Overriding by using three classes.**

**Program :**

**#include<iostream>**

**using namespace std;**

**class Div{**

**public:**

**Division(int a, int b)**

**{**

**cout << "Answer : " << a/b << endl;**

**}**

**};**

**class divided : public Div{**

**public:**

**Division(int a, int b)**

**{**

**cout << "Div : " << a/b << endl;**

**}**

**};**

**class Divid : public divided{**

**public:**

**Division(int a, int b)**

**{**

**cout << a << " & " << b << " Division : " << a/b << endl;**

**}**

**};**

**int main()**

**{**

**int a,b;**

**cout << "Enter A : ";**

**cin >> a;**

**cout << "Enter B : ";**

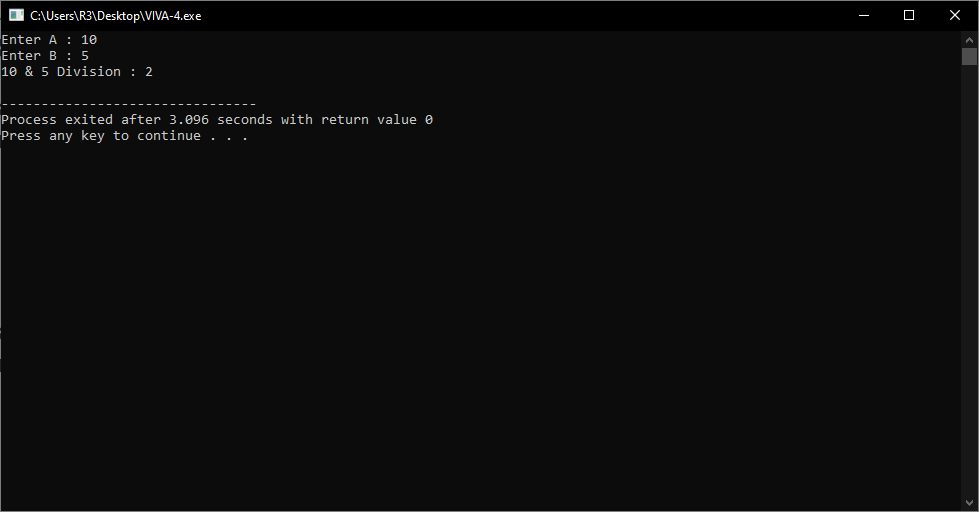
**cin >> b;**

**Divid d;**

**d.Division(a,b);**

**}**

**Output :**

****

**AIM :5 Write a Program to handle ambuigity error in Hybrid inheritance.**

**Program :**

**#include<iostream>**

**using namespace std;**

**class A**

**{**

**public:**

**virtual void classA()**

**{**

**cout<<"Class A"<<endl;**

**}**

**};**

**class B : virtual public A**

**{**

**public:**

**void classB()**

**{**

**cout<<"Class B"<<endl;**

**}**

**};**

**class C : virtual public A**

**{**

**public:**

**void classC()**

**{**

**cout<<"Class C"<<endl;**

**}**

**};**

**class D : public C,public B**

**{**

**public:**

**void classD()**

**{**

**cout<<"Class D"<<endl;**

**}**

**};**

**int main()**

**{**

**D d1;**

**d1.classD();**

**}**

**Output :**

