**Phase - 6 OOP Programs**

**Practical-2**

**Aim: A Businessman was bankrupted in a Scan with a minimal**

**amount left in a bank of ₹.18,000. After some months of hardwork**

**,he earned external amount of ₹.1,20,000.Now he might be goes to**

**the bank and do a deposit or withdraw some money as he wants.**

**Prepare a C++ solution for this scenario with all required validations and criterias.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class business

{

private:

char acc\_no1[100]="1234567890";

char pass1[100]="24102002";

char acc\_no[100], pass[100];

char name[100];

int amt,c,d;

public:

void businessman()

{

cout<<endl<<"=> Enter Name: ";

fflush(stdin);

gets(name);

cout<<"=> Enter Account Number: ";

fflush(stdin);

gets(acc\_no);

cout<<"=> Enter Password: ";

fflush(stdin);

gets(pass);

}

void List()

{

cout<<endl<<"\* [1] Deposite "<<endl;

cout<<"\* [2] Withdraw "<<endl;

cout<<"\* [0] Exit "<<endl;

}

void solution()

{

if(strcmp(acc\_no1,acc\_no)==0 && strcmp(pass1,pass)==0 )

{

cout<<"=> Enter Total Amount: ";

cin>>amt;

if(amt==18000)

{

do{

List();

cout<<"=> Enter Your Choice: ";

cin>>c;

if(c==1)

{

cout<<"- How many deposite:- ";

cin>>d;

amt+=d;

cout<<endl<<"- Your balance : "<<amt<<endl;

}

else if(c==2)

{

cout<<"- How many Withdraw:- ";

cin>>d;

amt-=d;

cout<<endl<<"- Your balance is: "<<amt<<endl;

}

else if(c!=0)

{

cout<<endl<<"- Please, Enter Valid Value...."<<endl;

}

}while(c!=0);

}

else

{

cout<<endl<<"- Sorry, You aren't able to open..."<<endl;

}

}

else

{

cout<<endl<<"- Please Enter Right Account no and Password...."<<endl;

}

}

};

int main()

{

business b1;

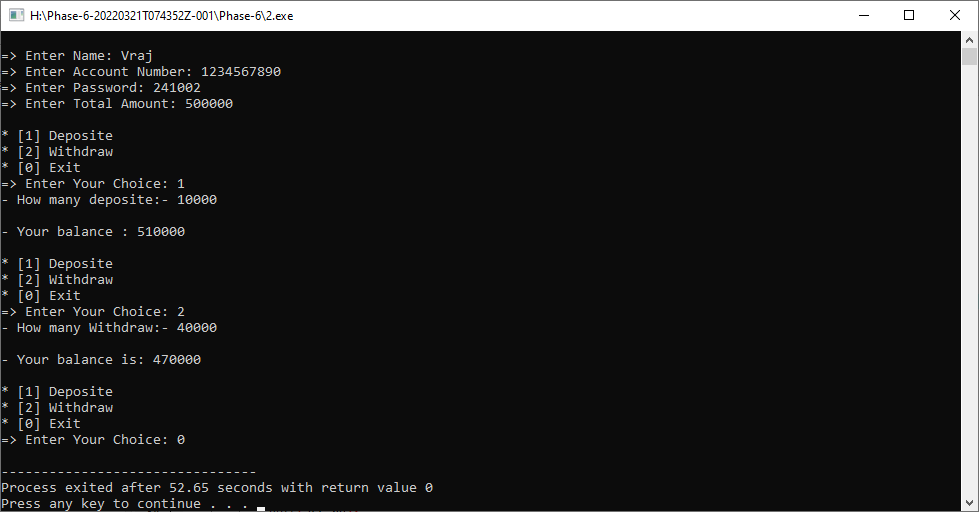
b1.businessman();

b1.solution();

return 0;

}

**Output:**

****

**Practical-3**

**Aim: An Auction is helding at Arizona for selling an old haunted house. For the reason, this is a haunted house,only three gigantic companies took a part in this Auction.Sell this haunted house to the highest bidder with count of three. Use C++ with all required criteria to build this type of Auction System.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Auction

{

private:

int a,b,c,choice;

public:

void sell()

{

do{

cout<<endl<<"=> Enter 1 Company Prize: ";

cin>>a;

cout<<endl<<"=> Enter 2 Company Prize: ";

cin>>b;

cout<<endl<<"=> Enter 3 Company Prize: ";

cin>>c;

if(a>b)

{

if(a>c)

{

cout<<endl<<" First Company Win Successfully...."<<endl;

}

else

{

cout<<endl<<" Third Company Win Successfully...."<<endl;

}

}

else

{

if(b>c)

{

cout<<endl<<" Second Company Win Successfully...."<<endl;

}

else

{

cout<<endl<<" Third Company Win Successfully...."<<endl;

}

}

cout<<endl<<"\* Press 1 for Continue.."<<endl;

cout<<"\* Press 0 for Exit.."<<endl;

cout<<endl<<" Enter Choice: ";

cin>>choice;

}while(choice!=0);

}

};

int main()

{

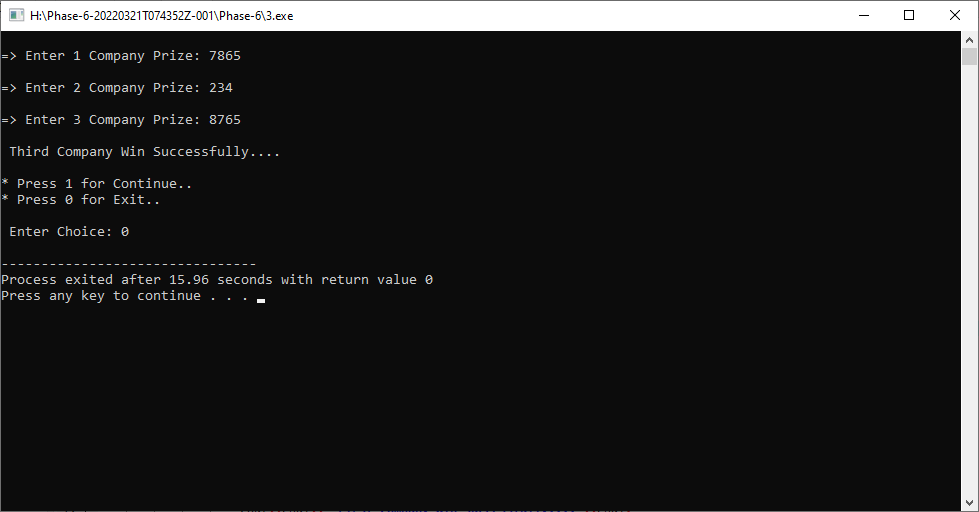
Auction a1;

a1.sell();

return 0;

}

**Output:**

****

**Practical-4**

**Aim: Build a C++ system which predict a total profit of a Cashew Company in Goa.If this company sells 1,23,500 piece of cashews in 1 month,then it generates total of ₹.78,000 in a month.Help this company by producing 10X more cashews in 3 months and display total revenue with increment percentage.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Cashew

{

private:

int c=123500, Rs=78000, S\_C , amt , p;

public:

void sell()

{

S\_C = c\*10;

amt = Rs\*10;

p = ((S\_C\*100)/c)/3;

}

void getdata()

{

sell();

cout<<endl<<"-> This Company should Producing "<<S\_C<<" Cashew."<<endl;

cout<<endl<<"-> Total Revenue with increment percentage is : "<<p<<"%"<<endl;

}

};

int main()

{

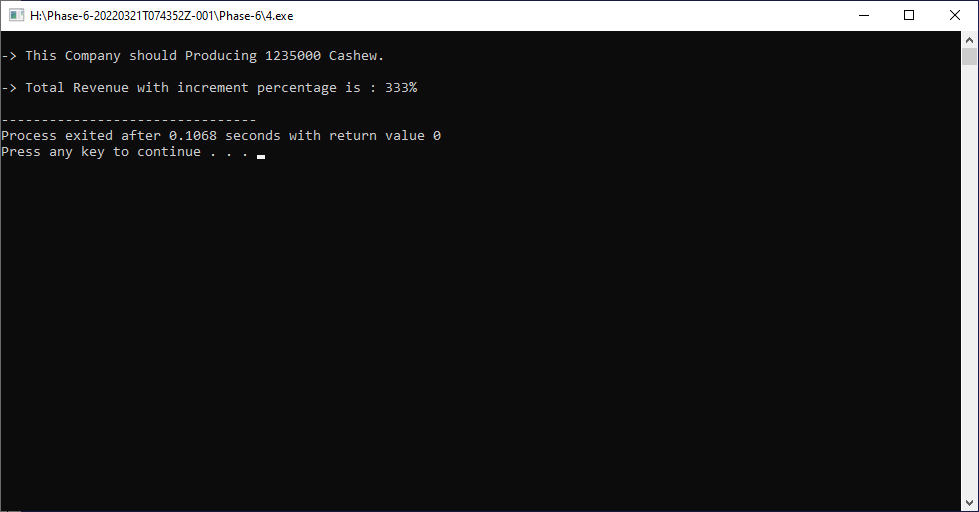
Cashew c1;

c1.getdata();

return 0;

}

**Output:**

****

**Practical-5**

**Aim: The two short sides of a right triangle are 6 cm and 13 cm.Find**

**the length of the third side using Pythagoras Theorem with help of C++.**

**Program:**

#include<iostream>

#include<string.h>

#include<math.h>

using namespace std;

class Sides

{

private :

int AB ; // AC^2 = AB^2 + BC^2

int BC ;

int AC ;

int p;

public :

void Side\_setData()

{

this->AB = 13;

this->BC = 6;

cout <<endl<<"-------------------------------------"<<endl;

cout <<"=> AC^2 = AB^2 + BC^2 :- "<<endl;

cout <<"-------------------------------------"<<endl;

cout <<endl<<"=> First Side (AB) : "<<this->AB;

cout <<endl<<"=> Second side (BC) : "<<this->BC;

cout <<endl<<endl<<"-------------------------------------";

}

void getData()

{

AC = (AB\*AB)+(BC\*BC);

p = sqrt(AC);

cout <<endl<<"=> Third Side (AC) : "<<p <<endl;

cout <<"-------------------------------------"<<endl;

}

};

int main()

{

Sides s1;

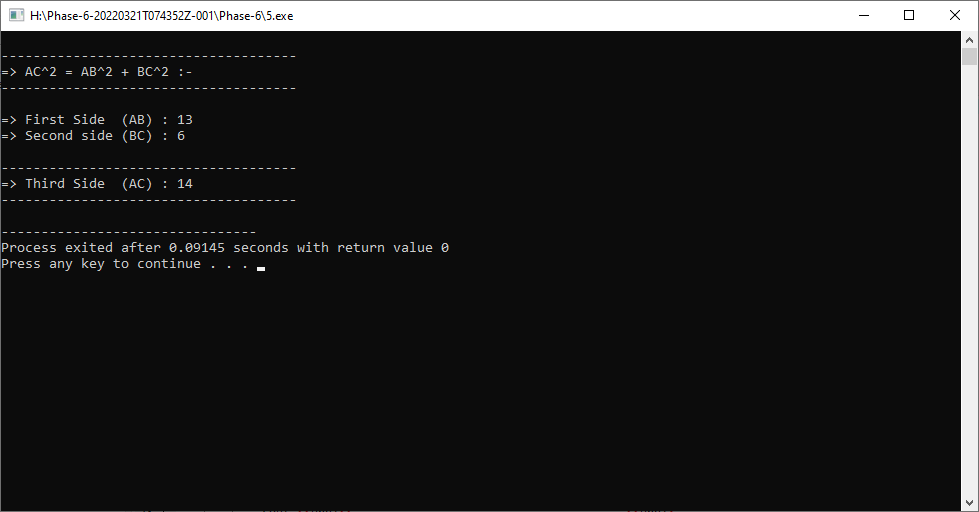
s1.Side\_setData();

s1.getData();

return 0;

}

**Output:**

****

**Practical-6**

**Aim: TA 26 m long rope is stretched from the top of a 13 m**

**tree to the ground. Find the distance between the tree and the**

**end of the rope on the ground.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Distance

{

private:

int a=26-13 ;

int b=13;

int c;

int d;

int temp=0;

public:

void getdata()

{

d=(a\*a)+(b\*b);

c = d/ 2;

while(c!=temp)

{

temp = c;

c = (d/temp + temp)/2;

}

cout<<endl<<"=> The distance between the tree and the end of the rope on the ground is: "<<c<<endl;

}

};

int main()

{

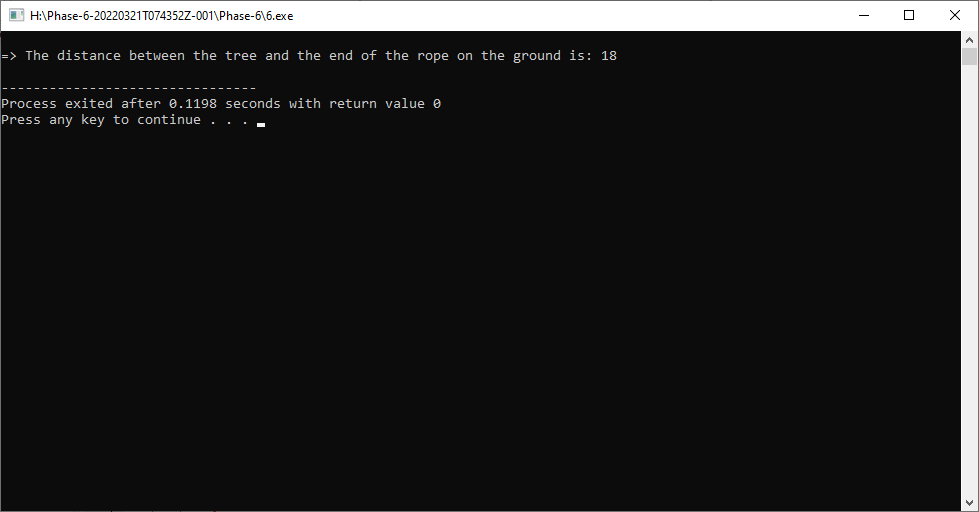
Distance d1;

d1.getdata();

return 0;

}

**Output:**

****

**Practical-7**

**Aim: Build a C++ system which helps a Mathematician to figure out**

**the type of a Triangle. Bases on Pythagoras’ theorem, find out if a tria-**

**ngle is: obtuse, right or acute.**

**Program:**

#include<iostream>

using namespace std;

class Square

{

private:

int a;

int b;

int c;

int sum;

public:

void S()

{

cout <<endl<< "=> Enter value of a :- "; cin >> this->a;

cout << "=> Enter value of b :- "; cin >> this->b;

cout << "=> Enter value of c :- "; cin >> this->c;

c= c\*c;

sum = (a\*a)+(b\*b);

if(c==sum)

{

cout <<endl<< "- Right....";

}

else if (c>sum)

{

cout <<endl<< "- Obtuse....";

}

else

{

cout <<endl<< "- Acute....";

}

}

};

int main()

{

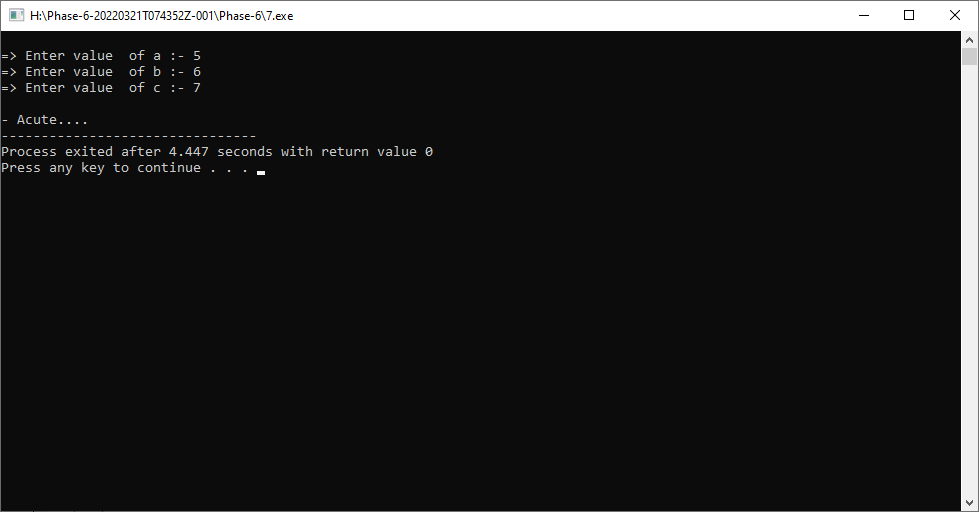
Square s1;

s1.S();

return 0;

}

**Output:**

****

**Practical-8**

**Aim: A 15 m fire-fighter’s ladder is leaning against the wall. If the**

**ground distance between the foot of the ladder and the wall is 7 m, \**

**what is the wall’s height?**

**Program:**

#include<iostream>

#include<string.h>

#include<math.h>

using namespace std;

class Distance

{

private:

int a = 15;

int b = 7;

int c;

int k;

public:

void getData()

{

c=(a\*a)+(b\*b);

k=sqrt(c);

cout <<endl<<"- Height of wall : "<<k;

}

};

int main()

{

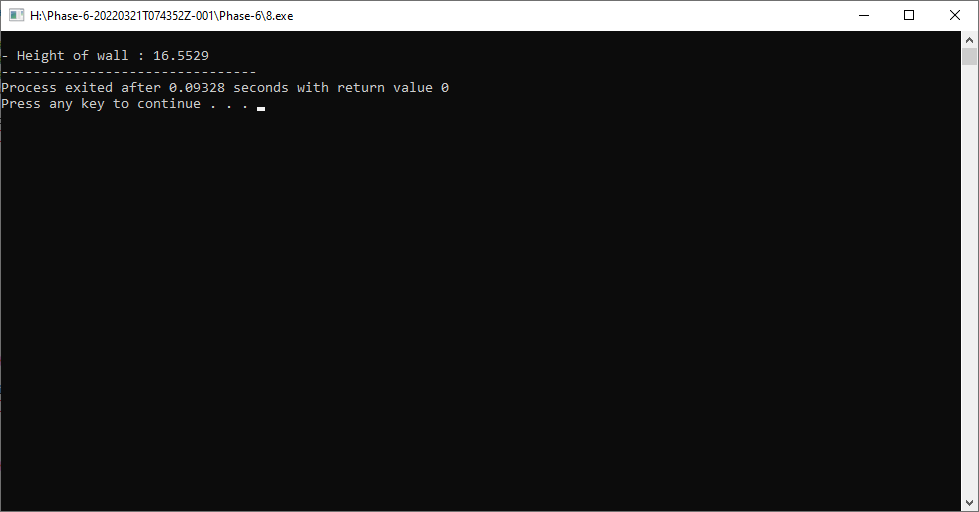
Distance d1;

d1.getData();

return 0;

}

**Output:**

****

**Practical-9**

**Aim: Design a GST Calculator in C++ to find total TAX on various typeof categorized items. Apply proper types of Indian GST TAX varients based on different types of Goods. GST have been divided into four GST rates – 5%,12%, 18%, and 28% by the GST Council.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class GST\_cal

{

private:

int price;

int gst;

int total\_price;

public :

void setData()

{

cout <<"......... \* GST Calculator \* ........."<<endl;

cout <<endl<<"=> Enter Price : ";

cin >>this->price;

}

void getData()

{

if(price<=500)

{

gst=(price\*5)/100;

}

else if(price>=500 && price<=1000)

{

gst=(price\*12)/100;

}

else if(price>=1000 && price<=2000)

{

gst=(price\*18)/100;

}

else

{

gst=(price\*28)/100;

}

total\_price = price+gst;

cout <<endl<<"=> Total GST price : "<<gst <<endl;

cout <<endl<<"---------------------------------------"<<endl;

cout <<"=> Total TAX : "<<total\_price;

cout <<endl<<"---------------------------------------"<<endl;

}

};

int main()

{

GST\_cal g1;

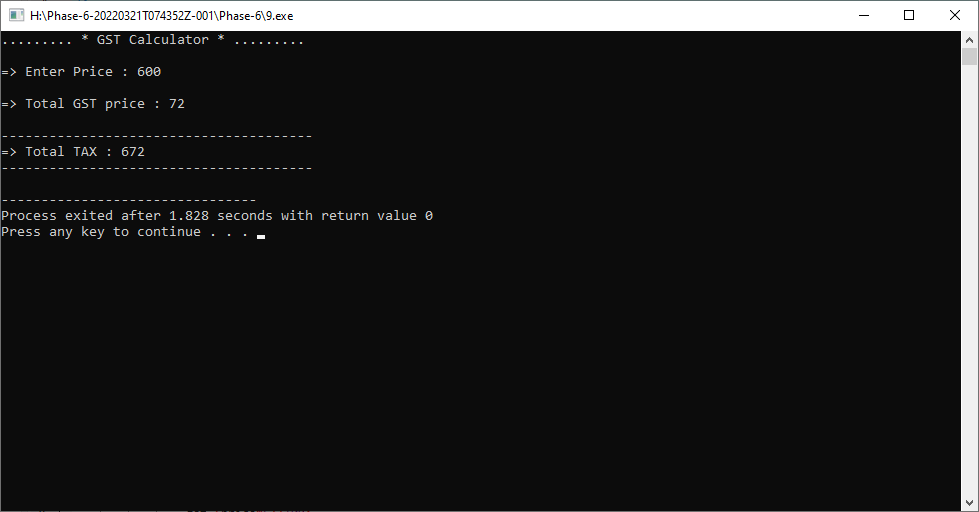
g1.setData();

g1.getData();

return 0;

}

**Output:**

****

**Practical-10**

**Aim: Develop a C++ solution by which a user can add/subtract/multiply/**

**divide two Complex numbers with help of Operator Overloading concept.**

**In context of math, a complex number contains two parts: a real part and**

**an imagenary part.**

**Program:**

**Output:**

**Practical-11**

**Aim: Build an Indian Regional Festival system in C++. User can enter**

**any date of current running year, and bases on that date display which**

**festival will be coming on that date.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Festival

{

private :

int date;

int month;

public :

void setData()

{

cout <<endl<<"=> Enter Month : ";

cin >>this->month;

cout <<"=> Enter Date : ";

cin >> this->date;

}

void getData()

{

cout <<endl<<"=> "<<date<<"/"<<month<<"/"<<"2022"<<endl;

if(date==14 && month==1)

{

cout <<endl<<"- Makar Sanskranti";

}

else if(date==17 && month==1)

{

cout <<endl<<"- Pongal";

}

else if(date==16 && month==2)

{

cout <<endl<<"- Basant panchami";

}

else if(date==1 && month==3)

{

cout <<endl<<"- Mahashivratri";

}

else if(date==17 && month==3)

{

cout <<endl<<"- Holika Dahan";

}

else if(date==18 && month==3)

{

cout <<endl<<"- Holi";

}

else if(date==16 && month==4)

{

cout <<endl<<"- Hanuman Jayanti";

}

else if(date==3 && month==5)

{

cout <<endl<<"- Akshaya Tritiya";

}

else if(date==2 && month==8)

{

cout <<endl<<"- Naga Panchami";

}

else if(date==31 && month==9)

{

cout <<endl<<"- Ganesh Chaturthi";

}

else if(date==3 && month==10)

{

cout <<endl<<"- Navratri";

}

else if(date==10 && month==10)

{

cout <<endl<<"- Dusshera";

}

else if(date==23 && month==10)

{

cout <<endl<<"- Dhanteras";

}

else if(date==24 && month==10)

{

cout <<endl<<"- Diwali";

}

else if(date==26 && month==10)

{

cout <<endl<<"- Bhai Dooj";

}

else

{

cout <<endl<<"- Invalid Choice....";

}

}

};

int main()

{

Festival f1;

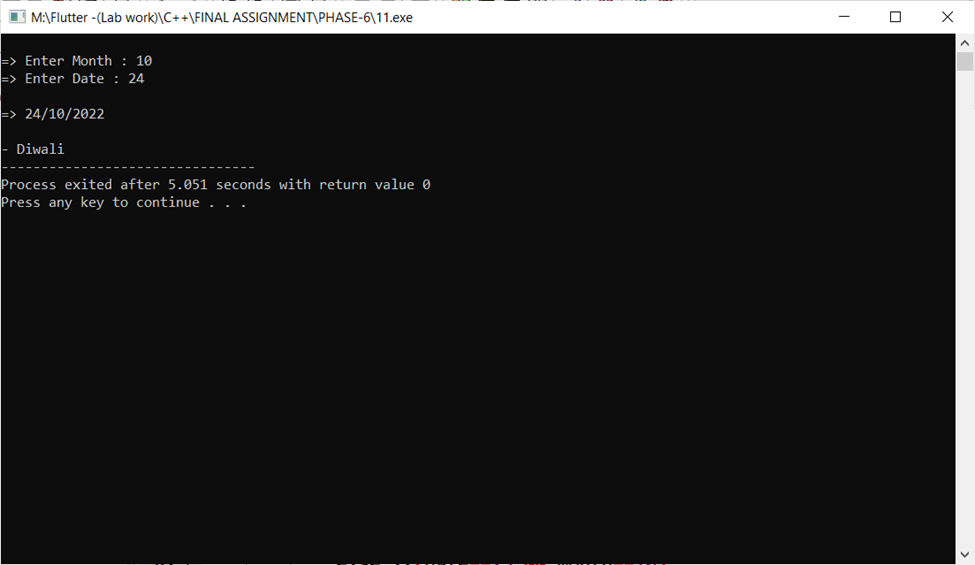
f1.setData();

f1.getData();

return 0;

}

**Output:**

****

**Practical-12**

**Aim: Prince wants to create a 24 Hr time convertor app in C++. In this app, user can provide any 24 Hr time he/she wants but output must be produced in 12 Hr format.**

**For example,**

**i/p: 15 Hr, 32 Minutes**

**o/p: 3:32 PM**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Time\_Convertor

{

private:

int hr;

int min;

public:

Time\_Convertor()

{

cout <<endl<<"=> Enter Hour : ";

cin >>this->hr;

cout <<endl<<"=> Enter Minute : ";

cin >>this->min;

}

void TC\_getData()

{

if(hr<=12)

{

cout <<endl<<"=> Time(12 hr) : "<<hr <<":" <<min <<endl;

}

else

{

cout <<endl<<"=> Time(12 hr) : "<<hr-12 <<":" <<min <<endl;

}

}

};

int main()

{

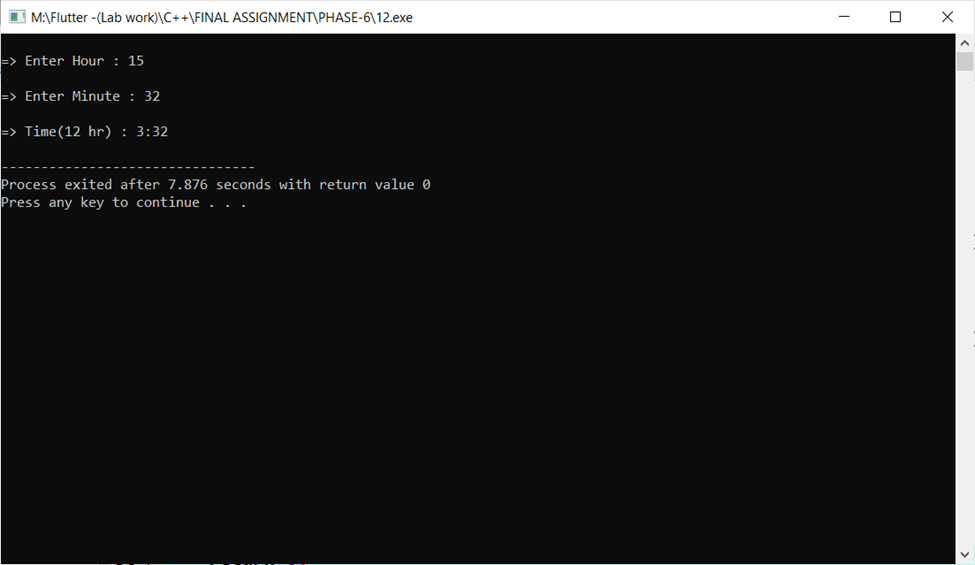
Time\_Convertor t;

t.TC\_getData();

return 0;

}

**Output:**

****

**Practical-13**

**Aim: Build a Counter App in C++ using OOP concept. Initially the counter meant to be set as a value 0 using constructor. By pressing UPArrow from keyboard, counter will be increment and by pressing DOWN Arrow, counter will be decrement. You can use ASCII value concept byachieving this type of functionality at the execution time of a Program.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Counter

{

private:

int n;

public:

void setData()

{

cout <<endl<<"=> Enter value of n : ";

cin >> this->n;

}

void List()

{

cout <<endl<<"(1) Press 1 for Increment "<<endl;

cout <<"(2) Press 2 for Decrement "<<endl;

}

void Increment()

{

n=n+1;

cout <<endl<<"-------------------------------"<<endl;

cout <<"=> Increment value : "<<n <<endl;

cout <<"-------------------------------"<<endl;

}

void Decrement()

{

n=n-1;

cout <<"-------------------------------"<<endl;

cout <<"=> Decrement value : "<<n <<endl;

cout <<"-------------------------------"<<endl;

}

};

int main()

{

Counter c1;

c1.setData();

int choice;

cout <<"-------------------------------"<<endl;

c1.List();

cout <<endl<<" => Enter your choice : ";

cin >>choice;

if(choice==1)

{

c1.Increment();

}

else if(choice==2)

{

c1.Decrement();

}

else

{

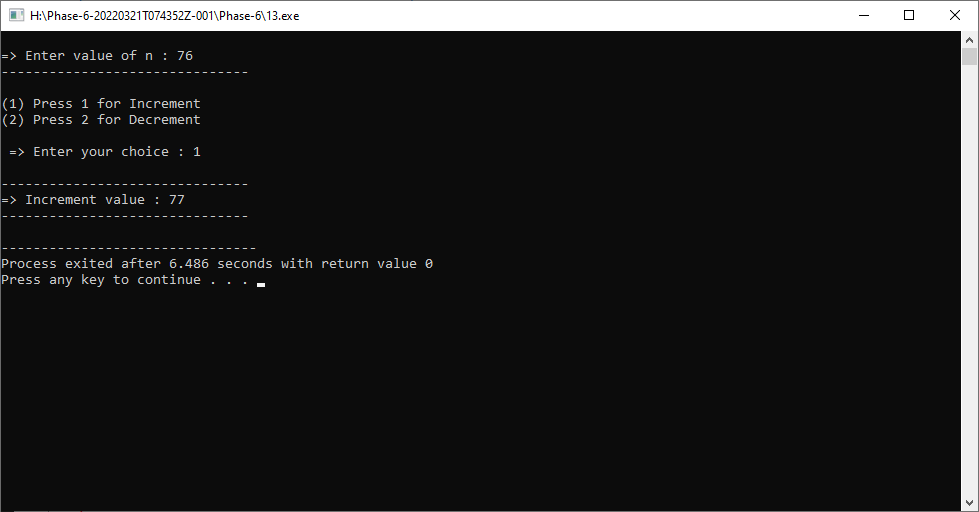
cout <<"-> Invalid choice..";

}

return 0;

}

**Output:**

****

**Practical-14**

**Aim: Calculate an Electricity Bill of a House of one month based**

**total units burned. Develop a C++ solution for this calculation.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Electricity\_Bill

{

private:

int unit;

int total;

int t;

public:

Electricity\_Bill()

{

cout <<endl<<"=> Enter Total Unit Usage : ";

cin >> unit;

}

void getData()

{

if(unit>0 && unit<=100)

{

cout <<endl<<"=> Your Bill Amount : ";

cout <<unit\*5;

}

else if(unit>100 && unit<=200)

{

cout <<endl<<"=> Your Bill amount : ";

cout <<(100\*5)+(unit-100)+7;

}

else if(unit>200 && unit<=300)

{

cout <<endl<<"=> Your Bill amount : ";

cout <<(100\*5)+(100\*7)+(unit-200)\*10;

}

else

{

cout <<endl<<"=> Your Bill amount : ";

cout <<"No value";

}

}

};

int main()

{

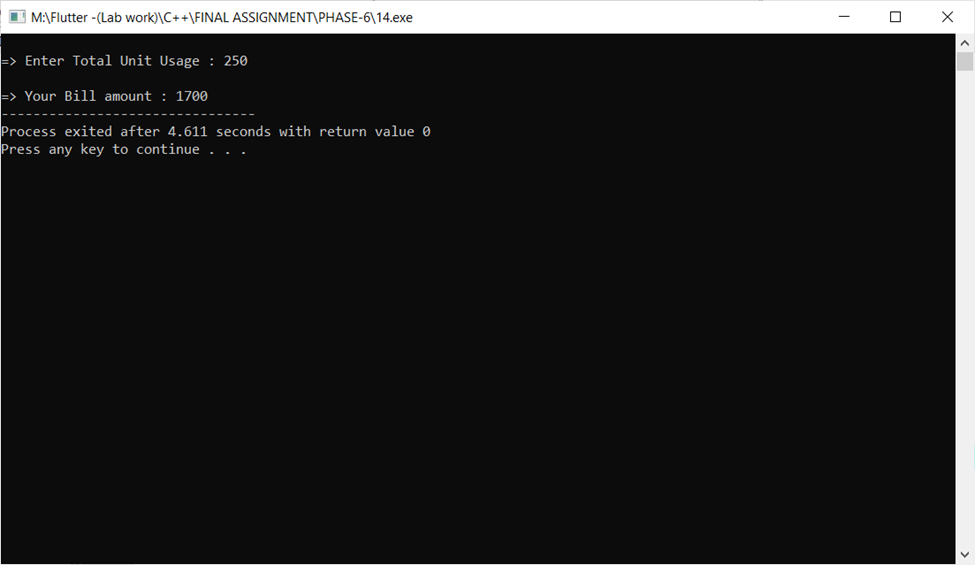
Electricity\_Bill e1;

e1.getData();

return 0;

}

**Output:**

****

**Practical-15**

**Aim: Calculate toal coast to apply a Solar Powered Panels for your Home Rooftop. Apply all types of government aid percentage to find reasonable coast.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Solar

{

private:

int cell;

int power;

int price;

int del\_chrg =1000;

int ord\_chrg =150;

int f\_c = 1500;

int total;

public:

Solar()

{

cout <<endl<<"=> Enter Number of cell : ";

cin >> cell;

cout <<"=> Enter power : ";

cin >> power;

}

void getData()

{

if(cell<=70)

{

price = 15000;

}

else if(cell<=150)

{

price=25000;

}

else

{

price=45000;

}

total=price+((price\*20)/100)+del\_chrg+ord\_chrg+f\_c;

cout<<endl<<"- Total Cost : " <<price <<endl;

cout<<"- Total GST : " <<(price\*20)/100<<endl;

cout<<"- Total Delivery Charge : "<<del\_chrg<<endl;

cout<<"- Total Other Charge : "<<ord\_chrg<<endl;

cout<<"- Total Fitting Charge : "<<f\_c<<endl<<endl;

cout<<"....................................."<<endl;

cout<<"- Total Cost : "<<total<<endl;

cout<<"....................................."<<endl;

}

};

int main()

{

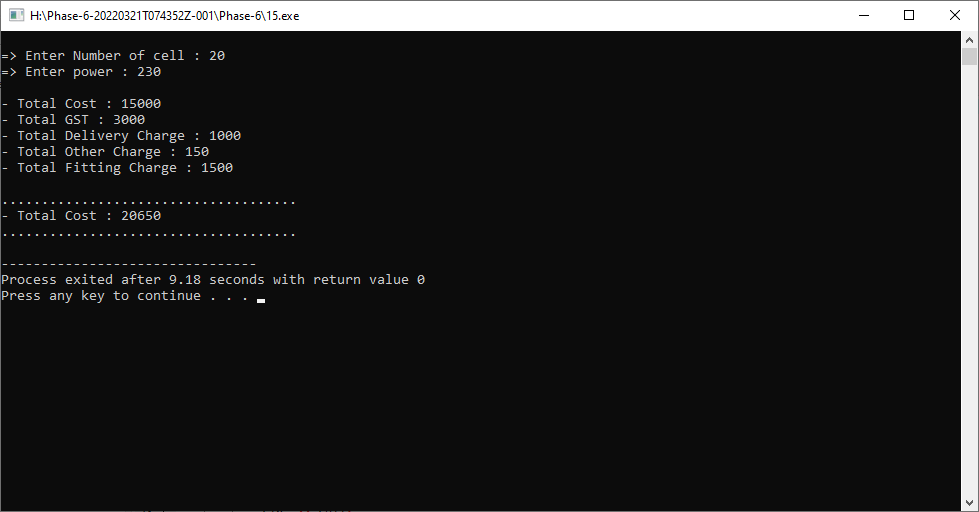
Solar s1;

s1.getData();

return 0;

}

**Output:**

****

**Practical-16**

**Aim: Find Volume of a Box using Parameterized Constructor**

**and figure out if that is odd or even number.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Box

{

private:

int volume;

public:

//Parameterized constructor

Box(int l , int b , int h)

{

volume = l\*b\*h;

cout <<endl <<"-> Volume of Box : "<<volume <<endl;

if(volume%2==0)

{

cout <<endl <<"-> This volume of box is even number ."<<endl;

}

else

{

cout <<endl <<"-> This volume of box is odd number ."<<endl;

}

}

};

int main()

{

int l ,b,h;

cout

cout <<"Enter Length : ";

cin >> l;

cout <<"Enter Breadth : ";

cin >> b;

cout <<"Enter Height : ";

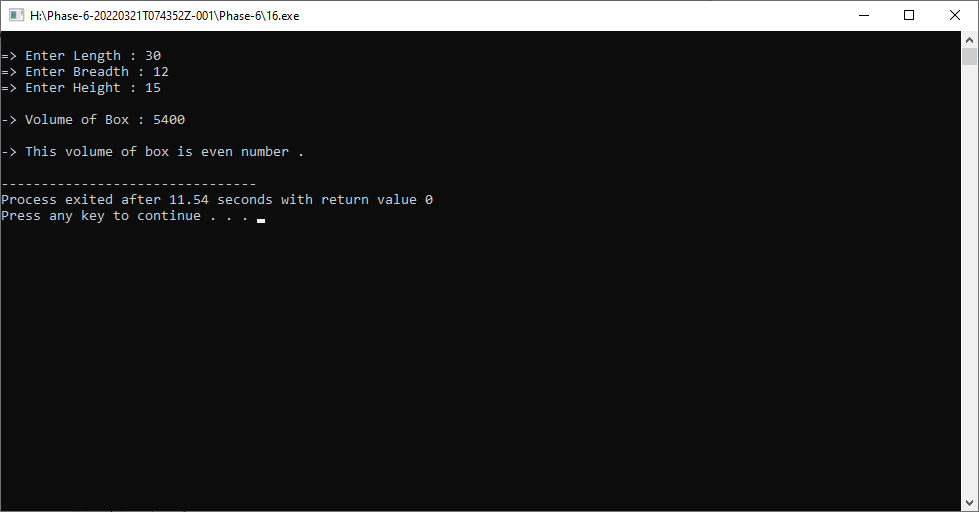
cin >> h;

Box b1(l,b,h);

return 0;

}

**Output:**

****

**Practical-17**

**Aim: By creating below mentioned inherited structure of classes, Assume suitable data and member methods for creating a Cricket scenario and listing score tables for top five players.**

**Program:**

**Output:**

**Practical-18**

**Aim: Help Ayush to perfome given operation:**

**a. Assume any number**

**b. Add 8 in that number**

**c. Multiply it with 3**

**d. Subtract 12 from it**

**e. Add another 5 into that**

**f. Add your birth year in it**

**g. Subtract current year from that**

**Finally display which number he get after performing all**

**above mentioned operations and find is it divisible by 7**

**or not.**

**Program:**

**Output:**

**Practical-19**

**Aim: Help a builder to build a house as same as**

**structurized as below:**

**Assume suitable data and member methods. You can add your**

**own functionalities to enhance this solution.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class House

{

public:

void HouseData()

{

cout <<" Welcome "<<endl;

cout <<" This is my house."<<endl;

}

};

class Kitchen : public House

{

public:

void KitchenData()

{

HouseData();

cout <<" There is a big kitchen in my house.";

}

};

class Bedroom : public House

{

public:

void BedroomData()

{

cout <<" There are four bedroom in my house."<<endl;

}

};

class Backyard : public House

{

public:

void BackyardData()

{

cout <<" This is backyard area in my house."<<endl;

}

};

class Dinning\_table : public Kitchen

{

public :

void Dinning\_tableData()

{

KitchenData();

cout <<endl<<" This is a Dinning table."<<endl;

}

};

class Bathroom : public Bedroom

{

public:

void BathroomData()

{

cout <<" This is a bathroom."<<endl;

BedroomData();

}

};

class Garage : public Backyard

{

public :

void GarageData()

{

BackyardData();

cout <<" This is a garage."<<endl;

}

};

int main()

{

Dinning\_table d1;

Bathroom b1;

Garage g1;

d1.Dinning\_tableData();

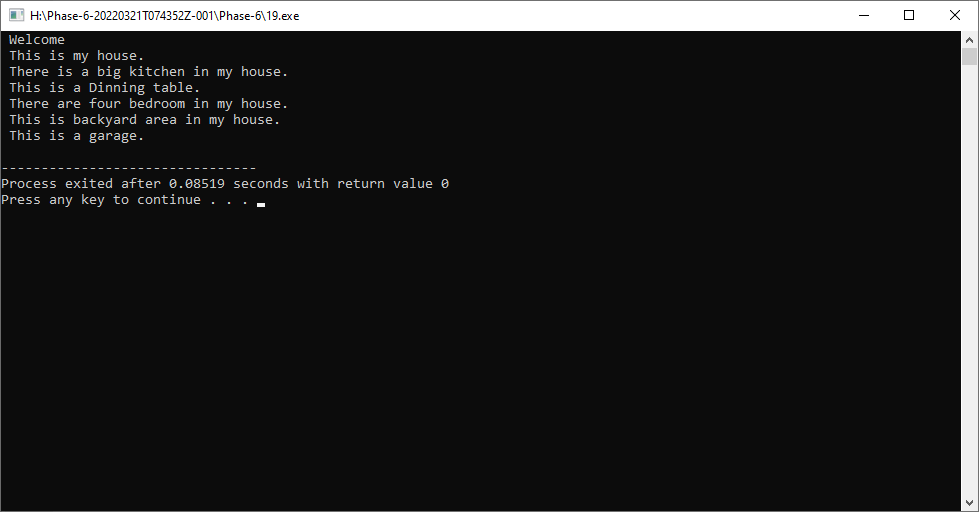
b1.BedroomData();

g1.GarageData();

return 0;

}

**Output:**

****

**Practical-20**

**Aim: A Higher Secondary School opens after COVID-19 crisis**

**and admission process will be starting for registration**

**of students. Help administration department for**

**registering student information such liker**

**stu\_i**

**stu\_nam**

**stu\_ag**

**stu\_cours**

**stu\_emai**

**stu\_college**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Student

{

private :

int stu\_id;

char stu\_name[100];

int stu\_age;

char stu\_course[100];

char stu\_email[100];

static char stu\_college[100];

public :

void Stu\_setData()

{

cout << endl<<"- Enter Student Id : ";

cin >> this->stu\_id;

cout <<"- Enter Student Name : ";

cin >> this->stu\_name;

cout <<"- Enter Student Age : ";

cin >> this->stu\_age;

cout <<"- Enter Student Course : ";

cin >> this->stu\_course;

cout <<"- Enter Student Email : ";

cin >> this->stu\_email;

}

void Stu\_getData()

{

cout <<endl<<"=> Enter Student Information : "<<endl<<endl

<<"- ID : "<<this->stu\_id <<endl

<<"- Name : "<<this->stu\_name <<endl

<<"- Age : "<<this->stu\_age <<endl

<<"- Course : "<<this->stu\_course <<endl

<<"- Email : "<<this->stu\_email <<endl

<<"- College : "<<this->stu\_college<<endl;

}

};

char Student :: stu\_college[100] = "BMU";

int main()

{

Student s1[100];

int i,n;

cout <<endl<<"=> How many Students : ";

cin >> n;

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

{

s1[i].Stu\_setData();

cout <<endl<<"----------------------------------"<<endl;

}

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

{

s1[i].Stu\_getData();

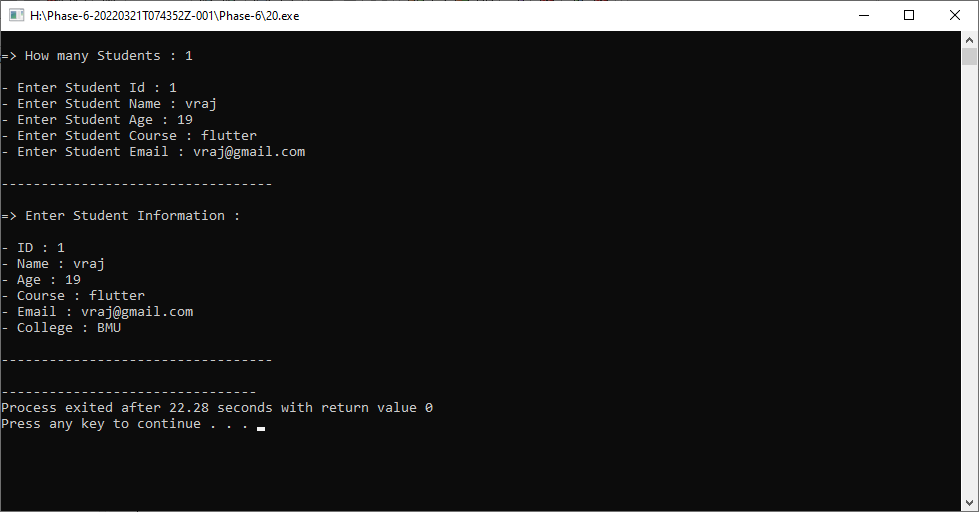
cout <<endl<<"----------------------------------"<<endl;

}

return 0;

}

**Output:**

****

**Practical-21**

**Aim: Build a C++ solution which returns array of all ASCII values of alphabets skipping 3 characters. Use concept of Constructors. After returning that array, find addition of that values and decide whether it is a odd or even number.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Alphabets

{

private :

char i;

int k = 0;

int s=0;

public :

Alphabets()

{

cout <<"=> All ASCII value of Alphabets : "<<endl<<endl;

for(i='a';i<='z';i+=3)

{

k= k+i;

cout <<"- Character "<<i <<" = " <<int(i) <<endl;

}

for(i='a';i<='z';i+=3)

{

s= s+i;

}

cout <<endl<<"=> Sum of a Character : "<<s <<endl;

if(k%2==0)

{

cout <<endl<<"=> Even Number...";

}

else

{

cout <<endl<<"=> Odd Number...";

}

}

};

int main()

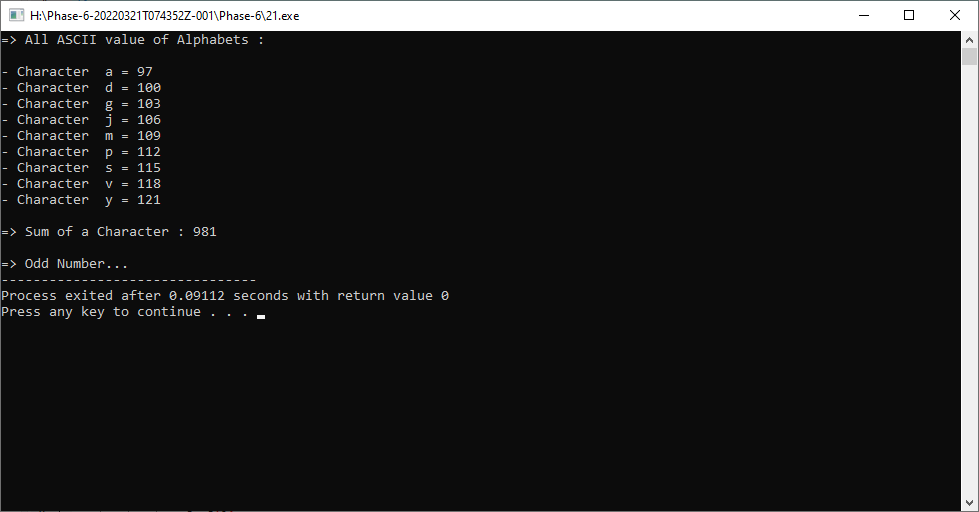
{

Alphabets();

return 0;

}

**Output:**

****

**Practical-22**

**Aim: A Global survey held to collect information about hotels all**

**around the world. Provide a C++ solution to create a class Hotel with**

**fields like**

**hotel\_id**

**hotel\_name**

**hotel\_type**

**hotel\_staff\_size**

**hotel\_room\_size**

**hotel\_establish\_year**

**hotel\_country**

**hotel\_rating\_type**

**hotel\_website**

**Illustrate the use of strict encapsulation with this**

**keyword.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Hotel

{

private:

int hotel\_id;

char hotel\_name[100];

char hotel\_type[100];

char hotel\_staff\_size[100];

int hotel\_room\_size;

int hotel\_establish\_year;

char hotel\_country[100];

int hotel\_rating\_type;

char hotel\_website[100];

public:

void Hotel\_setData()

{

cout <<endl<<"Enter Hotel Id : ";

cin >>this->hotel\_id;

cout <<"Enter Hotel Name : ";

cin >>this->hotel\_name;

cout <<"Enter Hotel Type : ";

cin >>this->hotel\_type;

cout <<"Enter Hotel Staff size : ";

cin >>this->hotel\_staff\_size;

cout <<"Enter Hotel Room size : ";

cin >>this->hotel\_room\_size;

cout <<"Enter Hotel Established year : ";

cin >>this->hotel\_establish\_year;

cout <<"Enter Hotel Country : ";

cin >>this->hotel\_country;

cout <<"Enter Hotel Rating type : ";

cin >>this->hotel\_rating\_type;

cout <<"Enter Hotel website : ";

cin >>this->hotel\_website;

}

void Hotel\_getData()

{

cout <<endl<<"------ -: Enter Hotel Details :- ------"<<endl

<<" ID : "<<this->hotel\_id <<endl

<<" Name : "<<this->hotel\_name <<endl

<<" Type : "<<this->hotel\_type <<endl

<<" Staff size : "<<this->hotel\_staff\_size <<endl

<<" Room size : "<<this->hotel\_room\_size <<endl

<<" Established year : "<<this->hotel\_establish\_year <<endl

<<" Country : "<<this->hotel\_country <<endl

<<" Rating type : "<<this->hotel\_rating\_type <<endl

<<" website : "<<this->hotel\_website <<endl;

}

};

int main()

{

Hotel h[100];

int i,n;

cout <<endl<<" How many Hotel information : ";

cin >>n;

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

{

h[i].Hotel\_setData();

}

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

{

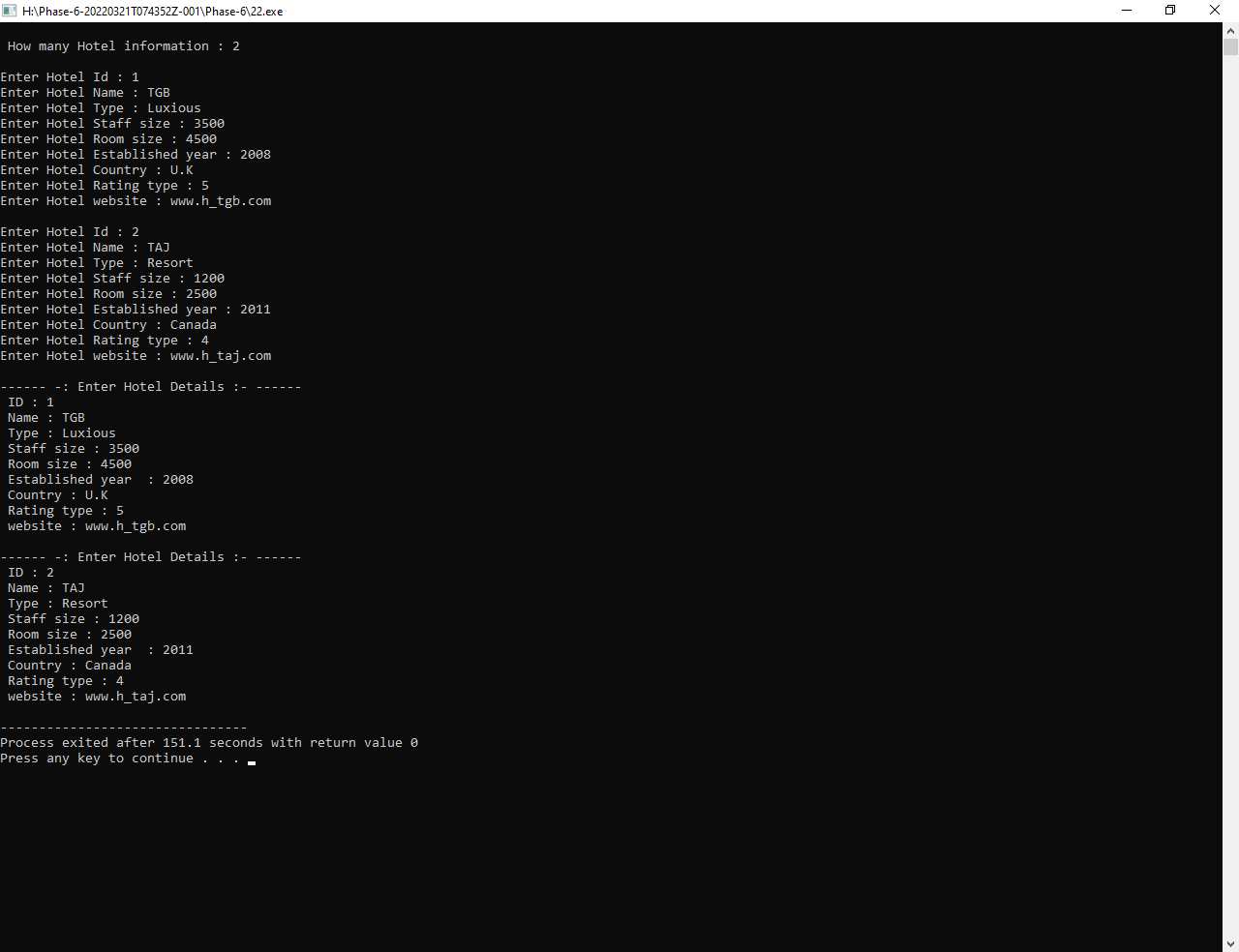
h[i].Hotel\_getData();

}

return 0;

}

**Output:**

****

**Practical-23**

**Aim: Jemin wants to create an automate system which compare two**

**given strings and it returns 1 if both strings are same and 0 otherwise.**

**Create a C++ system for helping Jemin using overloading concept.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class String

{

public:

int setdata(char a[], char b[])

{

if(strcmp(a,b)==0)

{

return 1;

}

else

{

return 0;

}

}

};

int main()

{

String s1;

char f[100], s[100];

int n;

cout<<endl<<"=> Enter First Message : ";

gets(f);

cout<<endl<<"=> Enter Second Message : ";

gets(s);

n=s1.setdata(f,s);

if(n==1)

{

cout<<endl<<"=> Both Message is Same..."<<endl;

}

else

{

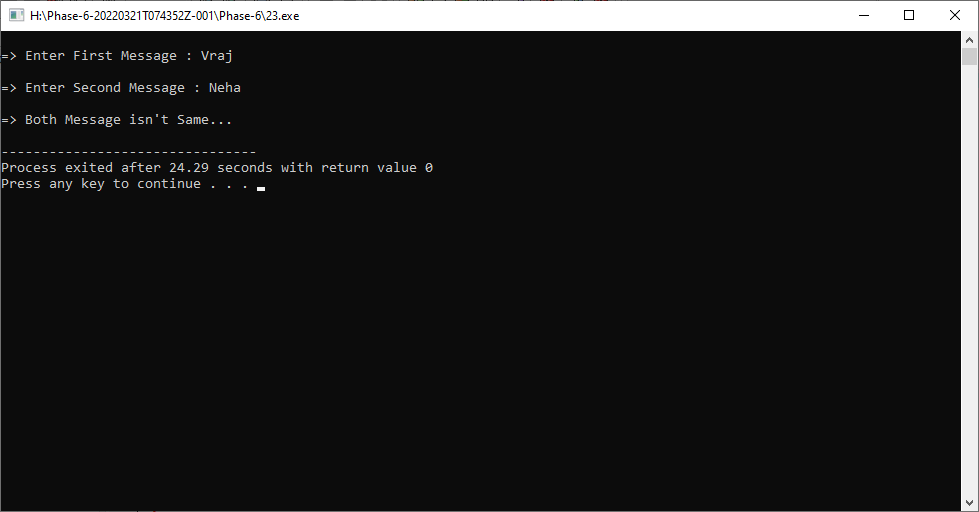
cout<<endl<<"=> Both Message isn't Same..."<<endl;

}

return 0;

}

**Output:**

****

**Practical-24**

**Aim: Design a swapping program using only constructors for**

**helping Devansh to gain passing marks in examination.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Swap\_Mark

{

public:

int a;

int b;

public:

Swap\_Mark()

{

cout <<"--------------------------------------"<<endl;

cout <<"=> After Swapping Passing Mark :- "<<endl;

cout <<"--------------------------------------"<<endl;

cout <<endl<<" \* Enter Original Mark : ";

cin >> a;

cout <<endl<<" \* Enter Passing Mark : ";

cin >> b;

// "<<endl <<" a : "<<a <<endl <<" b :"<<b <<endl;

a = a+b;

b = a-b;

a = a-b;

cout <<endl<<"--------------------------------------"<<endl;

cout <<"=> Before Swapping Passing Mark :- "<<endl;

cout <<"--------------------------------------"<<endl;

cout <<endl<<" \* Original Mark : "<<a;

cout <<endl<<endl<<" \* Passing Mark : "<<b;

}

};

int main()

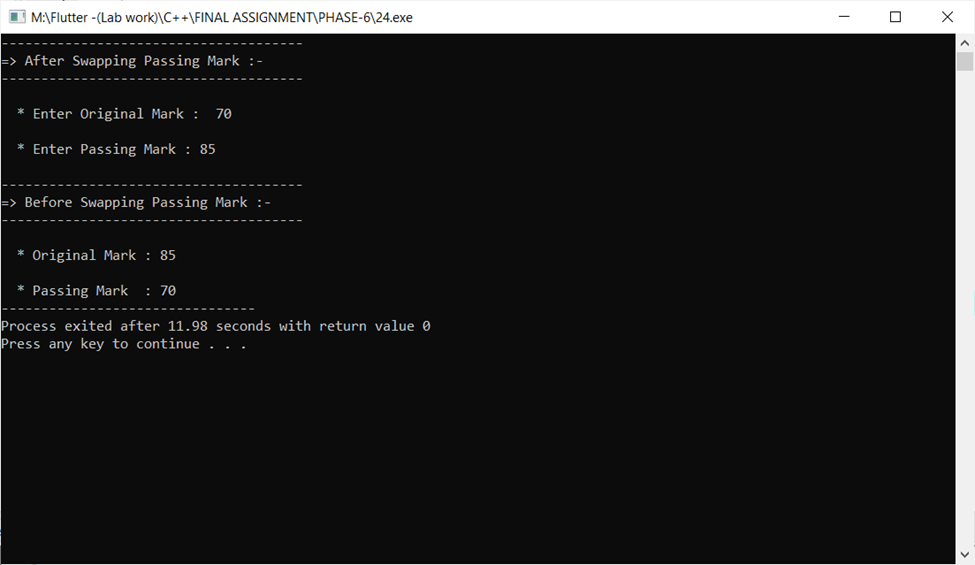
{

Swap\_Mark();

return 0;

}

**Output:**

****

**Practical-25**

**Aim: Create a C++ Base class Shape with a constructor for providing**

**values for width and height. Then define two derived classes Triangle**

**and Rectangle, that calculate the area of the shape area(). In the main,**

**define two objects: a triangle and a rectangle and then call the area()**

**function by this two objects.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Shape

{

public :

int width;

int height;

public :

void S1\_Data()

{

cout <<"- Enter Width : ";

cin >>this->width;

cout <<"- Enter Height : ";

cin >>this->height;

}

};

class Triangle : public Shape

{

public:

int area;

public :

void T\_Data()

{

S1\_Data();

area = (width\*height)/2;

cout <<endl<<"- Area of Triangle : "<<area <<endl;

}

};

class Rectangle : public Shape

{

public:

int area;

public :

void r\_Data()

{

S1\_Data();

area = (width\*height);

cout <<endl<<"- Area of Rectangle : "<<area <<endl<<endl;

}

};

int main()

{

Rectangle r1;

Triangle T1;

cout<<"-------------------------------------"<<endl;

cout <<"=> Find Area of Rectangle :- "<<endl;

cout<<"-------------------------------------"<<endl;

r1.r\_Data();

cout<<"-------------------------------------"<<endl;

cout <<"=> Find Area of Triangle :- "<<endl;

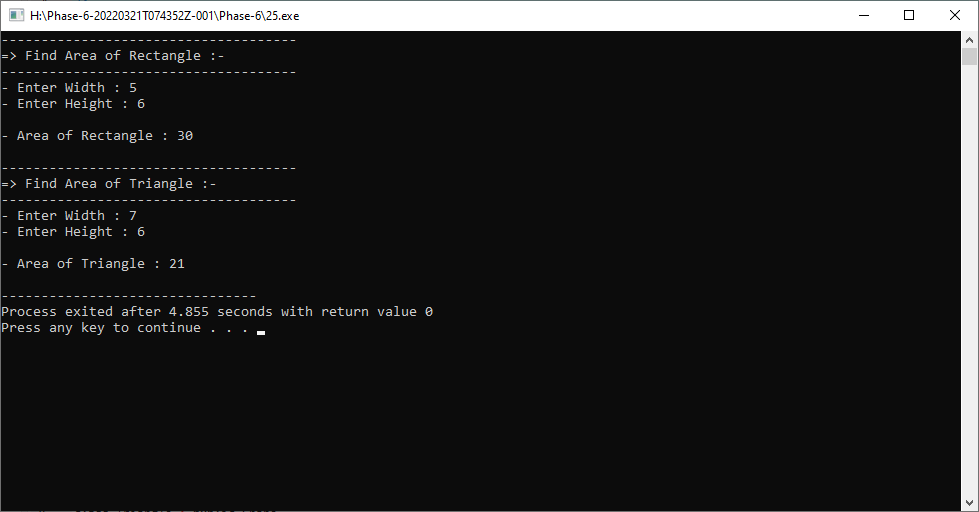
cout<<"-------------------------------------"<<endl;

T1.T\_Data();

return 0;

}

**Output:**

****

**Practical-26**

**Aim: By using Multilevel Inheritance, implement below mentioned**

**structure in C++ for employee records system.**

**Use proper implementation of Encapsulation, static keyword and Inheritance when needed.**

**Program:**

**Output:**

**Practical-27**

**Aim: Reserve Bank of India pre-defines a Rate of Interest (ROI) for**

**all banks across the Nation. But some private sector banks can apply**

**different ROI. Use inheritance and polymorphism concept to illustrate**

**this scenario.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class BOB

{

public :

void R\_of\_Int()

{

cout<<"------------------- \* BOB \* -------------------"<<endl;

cout <<endl<<"=> Pre-defines a Rate of Interest is 7.25% "<<endl<<endl;

}

};

class SBI : public BOB

{

public :

void R\_of\_Int()

{

cout<<"------------------- \* SBI \* -------------------"<<endl;

cout <<endl<<"=> Pre-defines a Rate of Interest is 6.25% "<<endl<<endl;

}

};

class RBI : public SBI

{

public :

void R\_of\_Int()

{

cout<<"------------------- \* RBI \* -------------------"<<endl;

cout <<endl<<"=> Pre-defines a Rate of Interest is 3.35% "<<endl<<endl;

BOB::R\_of\_Int();

SBI::R\_of\_Int();

}

};

int main()

{

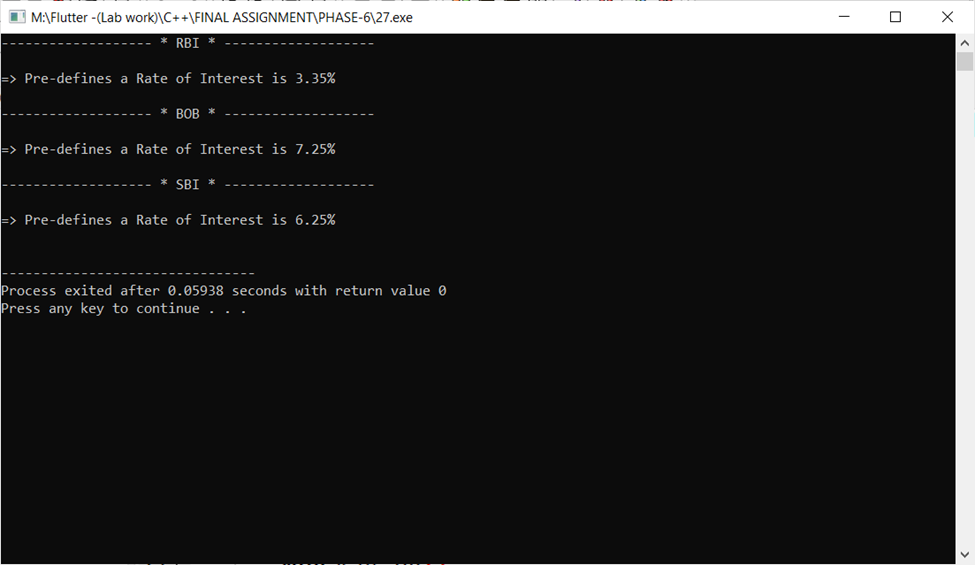
RBI r1;

r1.R\_of\_Int();

return 0;

}

**Output:**

****

**Practical-28**

**Aim: One character is common in both Marvel & DC universe named “Access”. Implement below mentioned structure using proper inheritance concept and with assumable fields and methods. You can create and use any type of methods and illustrations to enhance this program as per your preference.**

**Program:**

#include<iostream>

#include<string.h>

using namespace std;

class Universe

{

public:

void U\_getData()

{

cout <<endl<<"=> This is a universe :- "<<endl;

}

};

class Marvel : public Universe

{

public:

void M\_getdata()

{

cout <<"- This is a Marvel class of a universe ."<<endl;

}

};

class DC : public Universe

{

public:

void D\_getdata()

{

cout <<"- This is a DC class of a universe ."<<endl;

}

};

class MHeroes :public Marvel

{

public:

void MH\_getData()

{

cout <<"- This is a MHeroes class of a Marvel ."<<endl;

}

};

class DCHeroes :public DC

{

public:

void DC\_getData()

{

cout <<"- This is a DCHeroes class of a DC ."<<endl;

}

};

class Access : public MHeroes , public DCHeroes

{

public:

void AC\_getData()

{

Marvel::U\_getData();

M\_getdata();

D\_getdata();

cout <<endl<<"=> This is Access class :-"<<endl;

MH\_getData();

DC\_getData();

}

};

int main()

{

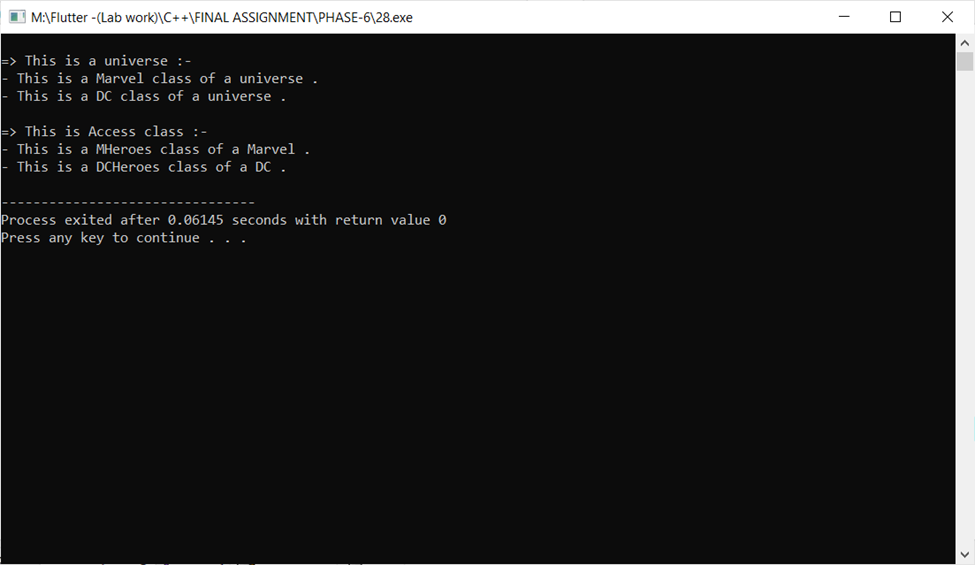
Access a1;

a1.AC\_getData();

return 0;

}

**Output:**

****