**Logo

Description automatically generated San Francisco Bay University**

**CS360L - Programming in C and C++ Lab**

**Lab Assignment #10**

**Due day: 08/02/2022**

**Instruction:**

1. **Push the answer sheets/source code to Github**
2. **Please follow the code style rule like programs on handout.**
3. **Overdue lab assignment submission can’t be accepted.**

**4. Take academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)**

1. Define a base class *shape* that can describe any simple shape such as a square, circle or equilateral triangle. The size of all these shapes can be reduced to a single dimension, such as area & perimeter of the shape based the all sizes

Define derived classes for each of the three shapes.

Create a virtual function in the base class that returns the area of each shape.

*Note: You will need to more precisely define what dimensions are stored in the base class. (Is the size in the base class for circle, the radius, or the diameter?)*

#include<iostream>

#include<conio.h>

#include<math.h>

using *namespace* std;

*class* shape{

*public:*

    virtual *void* area()=0; // pure virtual function

};

*class* circle: *public* shape{

*float* r; //r=radius

*public:*

*void* area(){

            cout<<"\nEnter radius : ";

            cin>>r;

            cout<<"\nArea of circle : "<<(3.146\*r\*r);

        }

};

*class* square: *public* shape{

*int* b;

*public:*

*void* area()

        {

        cout<<"\n\nEnter length of square: ";

        cin>>b;

        cout<<"\nArea of square : "<<b\*b;

        }

};

*class* equilateraltriangle: *public* shape{

*int* b;

*float* a;

*public:*

*void* area(){

        cout<<"\n\nEnter side of a Equilateral triangle : ";

        cin>>b;

        a=sqrt(3)/4\*(b\*b);

        cout<<"\nArea of triangle : "<<a;

    }

};

*int* main(*void*){

    circle c; // object created

    c.area();

    square r; // object created

    r.area();

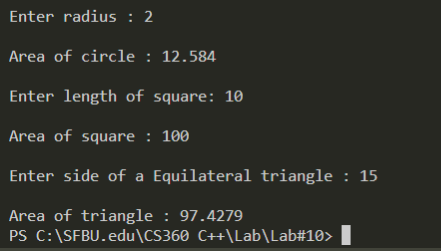
    equilateraltriangle t; // object created

    t.area();

    getch();

    return(0);

}



1. Write a base class called *person* that describes a person of either gender. Define two derived classes called *man* and *woman* that define gender specific items. Write pure virtual functions in the base class for operations that are common to both sexes yet are handled in different ways by each of them.

#include <iostream>

using *namespace* std;

*class* Person{

*public:*

        virtual *void* printName()=0;  //pure virtual function

        virtual *void* printGender()=0;

};

*class* Man : *public* Person{

*public:*

        string name;

*void* printName(){  //Print name of the person

            cout<<"Name: "<<name<<"\n";

        }

*void* printGender(){ //Print gender

            cout<<name<<" is a Male\n";

        }

};

*class* Woman : *public* Person{

*public:*

        string name;

*void* printName(){     //Print name of the person

            cout<<"Name: "<<name<<"\n";

        }

*void* printGender() {   //Print gender

            cout<<name<<" is a Female\n";

        }

};

*int* main(*void*) {

    Man m;

    Woman w;

    m.name="Christopher";

    m.printName();

    m.printGender();

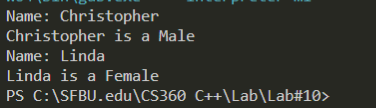
    w.name="Linda";

    w.printName();

    w.printGender();

    return 0;

}



1. Write a base class *number* that holds a single integer value and contains one member function, *print\_it*. Define three derived classes to print the value in hex, octal, and decimal.

#include <iostream>

using *namespace* std;

*class* number{

*public:*

*int* num;

        virtual *void* print\_it(*int* *num*)=0;  //pure virtual function

};

*class* hexVal : *public* number{

*public:*

*void* print\_it(*int* *num*){    //Printing Hexadecimal value

            cout<< "Hexadecimal value: "<<hex<<*num*<<"\n";

        }

};

*class* octVal : *public* number{

*public:*

*void* print\_it(*int* *num*){     //Printing Octal value

            cout<<"Octal value: "<< oct << *num* <<"\n";

        }

};

*class* decVal : *public* number{

*public:*

*void* print\_it(*int* *num*) {    //Printing Decimal value

            cout<< "Decimal value: "<<dec<<*num*<<"\n" ;

        }

};

*int* main(*void*) {

*int* numVal = 1000;

    cout << "Number: "<<numVal<<"\n";

    number \*h = new hexVal();    //Display Hexadecimal value

    h->print\_it(numVal);

    number \*o = new octVal();    //Display Octal value

    o->print\_it(numVal);

    number \*d = new decVal();    //Display Decimal value

    d->print\_it(numVal);

    return 0;

}

