National University of Computer and Emerging Sciences



Assignment 04

Object Oriented Programming

|  |  |
| --- | --- |
| Name | Muhammad Zain |
| Roll No. | 19F-0228 |
| Course Instructor | Dr Danish Shehzad |
| Semester | Spring 2020 |

Task 1

# Source Code:

#include<iostream>

using namespace std;

class Shape {

protected:

float length;

float width;

float height;

public:

Shape()

{

cout << "This is he constructor of base class" << endl;

}

virtual void setArea() = 0;

virtual void setPerimeter() = 0;

};

class Rectangle :public Shape {

public:

Rectangle() {

cout << "Enter the Length of Rectangle" << endl;

cin >> length;

cout << "Enter the Width of Rectangle" << endl;

cin >> width;

}

void setArea() {

cout << "The Area Of Rectangle is " << length\*width << " cm^2" << endl;

}

void setPerimeter() {

cout << "The Perimeter Of Rectangle is " << 2 \* (length + width) << " cm " << endl;

}

};

class Triangle :public Shape {

public:

Triangle() {

cout << "Enter the Base of Triangle" << endl;

cin >> length;

cout << "Enter the Height of Triangle" << endl;

cin >> height;

cout << "Enter the Width of Rectangle" << endl;

cin >> width;

}

void setArea() {

cout << "The Area Of Triangle is " << (length\*height) / 2 << " cm^2" << endl;

}

void setPerimeter() {

cout << "The Perimeter Of Triangle is " << (height + length + width) << " cm " << endl;

}

};

class Square :public Shape {

public:

Square() {

cout << "Enter the length ofone side of Square" << endl;

cin >> length;

}

void setArea() {

cout << "The Area Of Square is " << (length\*length) << " cm^2" << endl;

}

void setPerimeter() {

cout << "The Perimeter Of Square is " << 4 \* length << " cm " << endl;

}

};

int main()

{

int choice;

Shape \*shape\_pointer;

while (int x = 1)

{

cout << "If you want to calculate the area of Rectagle Press 1 " << endl;

cout << "If you want to calculate the area of Tritagle Press 2 " << endl;

cout << "If you want to calculate the area of Square Press 3 " << endl;

cout << "If you want to Exit.... Press 4 " << endl;

cout << endl;

cin >> choice;

system("cls");

//cout<<endl<<endl<<endl;

if (choice == 1)

{

Rectangle obj\_R;

shape\_pointer = &obj\_R;

shape\_pointer->setArea();

shape\_pointer->setPerimeter();

cout << endl << endl << endl;

}

else if (choice == 2)

{

Triangle obj\_T;

shape\_pointer = &obj\_T;

shape\_pointer->setArea();

shape\_pointer->setPerimeter();

cout << endl << endl << endl;

}

else if (choice == 3)

{

Square obj\_S;

shape\_pointer = &obj\_S;

shape\_pointer->setArea();

shape\_pointer->setPerimeter();

cout << endl << endl << endl;

}

else if (choice == 4)

{

break;

}

}

system("pause>0");

}

# Snip:

A screen shot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generatedA screen shot of a smart phone

Description automatically generatedA black sign with white text

Description automatically generated

Task 2:

# Source Code:

#include<iostream>

#include<string>

using namespace std;

class Publication {

protected:

string title;

float price;

public:

Publication()

{

title = "-";

price = 0;

}

void getData()

{

cout << "Input the Title of Publication" << endl;

cin >> title;

cout << "Input the Price of Publication" << endl;

cin >> price;

}

void putData()

{

cout << "The Title of Publication " << title << endl;

cout << "The Price of Publication " << price << " $" << endl;

}

};

class Book : public Publication {

protected:

int pageCount;

public:

Book()

{

pageCount = 0;

}

void getData()

{

Publication::getData();

cout << "Input the Number of Pages" << endl;

cin >> pageCount;

}

void putData()

{

Publication::putData();

cout << "The Total number of pages are " << pageCount << endl;

}

};

class Tape : public Publication {

protected:

float minute;

public:

Tape()

{

minute = 0;

}

void getData()

{

Publication::getData();

cout << "Input the minutes of playing track" << endl;

cin >> minute;

}

void putData()

{

Publication::putData();

cout << "The Total minutes of playing track is " << minute << endl;

}

};

int main()

{

int choice;

while (int x = 1)

{

cout << "Press 1 to check Publication Class (BASE CLASS)" << endl;

cout << "Press 2 to check Book Class (DERRIVED CLASS)" << endl;

cout << "Press 3 to check Tape Class (DERRIVED CLASS)" << endl;

cout << "Press 4 to Exit" << endl;

cout << endl;

cin >> choice;

system("cls");

if (choice == 1)

{

Publication Obj\_P;

Obj\_P.getData();

cout << endl << endl;

Obj\_P.putData();

cout << endl << endl << endl;

}

else if (choice == 2)

{

Book Obj\_B;

Obj\_B.getData();

cout << endl << endl;

Obj\_B.putData();

}

else if (choice == 3)

{

Tape Obj\_T;

Obj\_T.getData();

cout << endl << endl;

Obj\_T.putData();

cout << endl << endl << endl;

}

else if (choice == 4)

{

break;

}

}

system("Pause>0");

}

# Snips:

A close up of a black background

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screen shot of a smart phone

Description automatically generatedA screen shot of a computer

Description automatically generated

Task 3:

# Source Code:

#include<iostream>

#include<string>

#include<iomanip>

using namespace std;

class Mobile {

private:

string companyName[7] = { "Apple ","Samsung","Huawei ","Vivo ","Oppo ","OnePlus","Qmobile" };

double version[7] = { 12.05,8.75,2.58,9.05,2.89,3.01,11.9 }; //OS ya andriod version

int storage[7] = { 128,32,64,128,128,64,32 };

int camera[7] = { 20,18,16,15,12,10,5 };

float price[7] = { 250000,120000,50000,40000,450000,150000,20000 };

int warranty[7] = { 2,2,3,2,1,1,2 };

public:

Mobile()

{

cout << "This is Mobile class constructor " << endl;

}

friend void Camera(Mobile cam);

friend void Storage(Mobile cap);

friend void Versions(Mobile ver);

friend void Prices(Mobile rs);

void showPhones()

{

cout << "\t\t\t\tThese are the Smart phones available Now" << endl << endl << endl;

cout << "Company Name" << setw(20) << " Andriod or Os Version" << setw(30) << "Storage Capacity";

cout << setw(27) << "Camera" << setw(31) << "Warranty" << setw(15) << "Price" << endl << endl;

for (int i = 0; i<7; i++)

{

cout << companyName[i] << setfill(' ') << setw(20) << version[i] << " Fast Varriant " << setfill(' ') << setw(20) << storage[i] << " Gigabytes ";

cout << setfill(' ') << setw(20) << camera[i] << " Megapixel " << setfill(' ') << setw(20) << warranty[i] << " Years " << setw(15) << price[i] << " $ " << endl;

}

}

};

void Camera(Mobile cam)

{

int temp;

for (int i = 0; i<6; i++)

{

for (int j = 0; j<6; j++)

{

if (cam.camera[j]<cam.camera[j + 1])

{

temp = cam.camera[j];

cam.camera[j] = cam.camera[j + 1];

cam.camera[j + 1] = temp;

}

}

}

cout << "Camera Rankings of Phones are: " << endl << endl;

cout << "NO " << " Cameras" << endl;

for (int i = 0; i<7; i++)

{

cout << i + 1 << " " << cam.camera[i] << " megaPexil" << endl;

}

cout << endl << "If you want to choose the phone with best Camera Strength Choose the 1st one" << endl;

}

void Prices(Mobile rs)

{

int temp;

for (int i = 0; i<6; i++)

{

for (int j = 0; j<6; j++)

{

if (rs.price[j]>rs.price[j + 1])

{

temp = rs.price[j];

rs.price[j] = rs.price[j + 1];

rs.price[j + 1] = temp;

}

}

}

cout << "Phones According to Prices low to high " << endl << endl;

cout << "NO " << " Prices" << endl;

for (int i = 0; i<7; i++)

{

cout << i + 1 << " " << rs.price[i] << "$" << endl;

}

cout << endl << "If you want Purchase the phone with low price you can choose 1st one " << endl;

cout << endl << "If you want to choose the phone with high price you can choose the last one" << endl;

}

void Storage(Mobile cap)

{

int temp;

for (int i = 0; i<6; i++)

{

for (int j = 0; j<6; j++)

{

if (cap.storage[j]<cap.storage[j + 1])

{

temp = cap.storage[j];

cap.storage[j] = cap.storage[j + 1];

cap.storage[j + 1] = temp;

}

}

}

cout << "Phones According to storage from high to low " << endl << endl;

cout << "NO " << " storage" << endl;

for (int i = 0; i<7; i++)

{

cout << i + 1 << " " << cap.storage[i] << " giga Bytes" << endl;

}

cout << endl << "You can choose the 1st one because its STORAGE CAPACITY is maximum" << endl;

}

void Versions(Mobile ver)

{

int temp;

for (int i = 0; i<6; i++)

{

for (int j = 0; j<6; j++)

{

if (ver.version[j]<ver.version[j + 1])

{

temp = ver.version[j];

ver.version[j] = ver.version[j + 1];

ver.version[j + 1] = temp;

}

}

}

cout << endl << "Phones According to versions from high to low " << endl << endl;

cout << "NO " << " versions" << endl;

for (int i = 0; i<7; i++)

{

cout << i + 1 << " " << ver.version[i] << endl;

}

cout << endl << "You can choose 1st one beacuse its version is latest" << endl;

}

int main()

{

int temp;

Mobile compare;

compare.showPhones();

while (int x = 1)

{

cout << endl << endl << endl;

cout << "Press 1: If you want to compare cameras of all phones " << endl;

cout << "Press 2: If you want to compare storage capacities" << endl;

cout << "Press 3: If you want to compare Prices of varieants" << endl;

cout << "Press 4: If you want to compare Versions " << endl;

cout << "Press 5: If you want to EXIT" << endl;

cout << endl;

cin >> temp;

system("cls");

if (temp == 1)

{

Camera(compare);

}

else if (temp == 2)

{

Storage(compare);

}

else if (temp == 3)

{

Prices(compare);

}

else if (temp == 4)

{

Versions(compare);

}

else if (temp == 5)

{

break;

}

}

system("pause>0");

}

# Snip:

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screen shot of a social media post

Description automatically generatedA picture containing bird

Description automatically generated

Task 4:

# Source code:

#include<iostream>

#include<string>

#include<ctime>

#include<cstdlib>

using namespace std;

class Library;//forward declaration

class administration;//forward decleration

class student {

friend class Library;

friend class administration;

private:

string Name;

string Rollno;

public:

student()

{

Name = "-";

Rollno = "-";

}

void getData()

{

int temp;

cout << "Enter your Name" << endl;

cin >> Name;

cout << "Enter your Roll no" << endl;

cin >> Rollno;

}

};

class Library {

friend class student;

friend class administration;

private:

string book;

float bill;

student xyz;

public:

Library()

{

book = "-";

bill = rand() % 5000;

}

void getData()

{

int temp = 0;

cout << "Please Enter the Data to issue book " << endl << endl << endl;

cout << "Enter the name of the book you want " << endl;

cin >> book;

cout << "What is your name ?" << endl;

cin >> xyz.Name;

cout << "What is your Roll no ?" << endl;

cin >> xyz.Rollno;

cout << endl << endl << "Thankyou! BOOK has been issued" << endl;

}

};

class administration {

friend class student;

friend class library;

private:

float budget;

int reg;

student abc;

Library xyz;

public:

administration()

{

budget = rand() % 50000;

reg = rand() % 1000;

}

void getData()

{

int temp;

cout << "If you want to access library bill Press 1 " << endl;

cout << "If you want to access student data Press 2 " << endl;

cout << "If you want to check all of budget Press 3 " << endl;

cin >> temp;

if (temp == 1)

{

cout << "The Bill of the library is " << xyz.bill << " $" << endl;

}

else if (temp == 2)

{

cout << "Enter the name of student " << endl;

cin >> abc.Name;

cout << "Enter the roll number of the student " << endl;

cin >> abc.Rollno;

}

else if (temp == 3)

{

cout << "The Budget is " << budget << " $" << endl;

}

else

cout << "Invalid Input" << endl;

}

};

int main()

{

int temp;

srand(time(0));

while (int x = 1)

{

cout << "Press 1:If you want to access Student Class" << endl;

cout << "Press 2:If you want to access Student Class via Library Class" << endl;

cout << "Press 3:If you want to access Student Class and Library Class via Admin Class" << endl;

cout << "Press 4:If you want to Exit" << endl;

cin >> temp;

system("cls");

if (temp == 1)

{

student obj;

obj.getData();

cout << endl << endl;

}

else if (temp == 2)

{

Library obj;

obj.getData();

cout << endl << endl;

}

else if (temp == 3)

{

administration obj;

obj.getData();

cout << endl << endl;

}

else if (temp == 4)

{

break;

}

}

system("pause>0");

}

# Snip:

A screenshot of a computer

Description automatically generated

A close up of a logo

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A close up of a screen

Description automatically generatedA close up of a logo

Description automatically generated

A black and red text

Description automatically generated

Task 5:

# Source Code:

#include<iostream>

#include<string>

#include<iomanip>

using namespace std;

class Employee {

protected:

string First\_Name;

string Last\_Name;

int ISSN;

public:

Employee()

{

First\_Name = "-";

Last\_Name = "-";

ISSN = 0;

}

void setFirstname()

{

cout << "Enter First Name: ";

cin >> First\_Name;

while (First\_Name == "")

{

cin >> First\_Name;

}

}

void getFirstname()

{

cout << First\_Name;

}

void setLastname()

{

cout << "Enter Last Name: ";

cin >> Last\_Name;

while (Last\_Name == "")

{

cin >> Last\_Name;

}

}

void getLastname()

{

cout << Last\_Name;

}

void setISSN() {

cout << "Enter the International Standard Serial Number (ISSN) " << endl;

cin >> ISSN;

while (ISSN == 0)

{

cin >> ISSN;

}

}

int getISSN()

{

return ISSN;

}

virtual void display()

{

cout << endl << "First Name : "; getFirstname();

cout << endl << "Last Name : "; getLastname();

cout << endl << "ISSN : "; getISSN();

}

};

class HourlyEmployee :public Employee {

protected:

float grossPay;

float netPay;

float tax;

float hoursWorked;

float rate;

public:

HourlyEmployee()

{

Employee::setFirstname();

Employee::setLastname();

Employee::setISSN();

cout << " Enter the Hours Worked" << endl;

cin >> hoursWorked;

cout << "Enter the Rate per Hour" << endl;

cin >> rate;

cout << " 5% of tax from your pay as per the Company Policy" << endl;

tax = 5;

}

void setGrosspay() {

grossPay = hoursWorked\*rate;

}

void getGrossPay()

{

cout << "Your Gross pay is " << grossPay << " $ " << endl;

}

void setNetpay()

{

netPay = (grossPay\*tax) / 100;

}

void getNetpay()

{

cout << "Your Net pay is " << netPay << " $ " << endl;

}

void display() {

Employee::display();

getGrossPay();

getNetpay();

}

};

class SalariedEmployee :public Employee {

protected:

float MonthlygrossPay;

float MonthlynetPay;

float GpFund;

float daysWorked;

float earning;

public:

SalariedEmployee()

{

Employee::setFirstname();

Employee::setLastname();

Employee::setISSN();

cout << " Enter the Hours Worked" << endl;

cin >> daysWorked;

cout << "Enter the earning of One Day" << endl;

cin >> earning;

cout << " 10% GP fund is deducted" << endl;

GpFund = 10;

}

void setMonthlygrossPay()

{

MonthlygrossPay = daysWorked\*earning;

}

void getMonthlygrossPay()

{

cout << "Your Monthly Gross pay is " << MonthlygrossPay << " $ " << endl;

}

void setMonthlynetPay()

{

MonthlynetPay = (MonthlygrossPay\*GpFund) / 100;

}

void getMonthlynetPay()

{

cout << "Your Monthly pay is " << MonthlynetPay << " $ " << endl;

}

void display() {

Employee::display();

getMonthlygrossPay();

getMonthlynetPay();

}

};

int main()

{

int option;

do {

cout << "Press 1 for Hourly Employee Class" << endl;

cout << "Press 2 for Salaried Employee Class" << endl;

cout << "Press 3 to exit" << endl;

cin >> option;

system("cls");

if (option == 1)

{

HourlyEmployee obj1;

obj1.setGrosspay();

obj1.setNetpay();

obj1.display();

cout << endl << endl << endl;

}

else if (option == 2)

{

SalariedEmployee obj2;

obj2.setMonthlygrossPay();

obj2.setMonthlynetPay();

obj2.display();

cout << endl << endl << endl;

}

} while (option != 3);

system("pause>0");

}

# Snip:

A screen shot showing a black background

Description automatically generated

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated