OOP ASSIGNMENT

Name: Abdul Haseeb Arif

Reg no.: 2023-BS-AI-033

* Program No.1:

// File: Program no.1.cpp

// Date: 22-05-2024

// Name: Abdul Haseeb Arif

// Registration No: 2023-BS-AI-033

// Imagine a publishing company that markets both book and audiocasseƩe versions of its works. Create a

//class publicaƟon that stores the Ɵtle (a string) and price (type float) of a publicaƟon. From this class

//derive two classes: book, which adds a page count (type int), and tape, which adds a playing Ɵme in

//minutes (type float). Each of these three classes should have a getdata() funcƟon to get its data from the

//user at the keyboard, and a putdata() funcƟon to display its data. Write a main() program to test the

//book and tape classes by creaƟng instances of them, asking the user to fill in data with getdata(), and

//then displaying the data with putdata()

#include <iostream>

#include <string>

using namespace std;

class company {

public :

string title ;

float price ;

void getData1()

{

cout<<"enter title of the publication : ";

cin>>title;

cout<<"enter the price of the publification : ";

cin>>price;

}

void display1()

{

cout<<"the title is : "<<title<<endl;

cout<<"the price of publication is as : "<<price<<endl;

}

};

class book : public company {

public:

int pageCount;

void getData2()

{

cout<<"enter the amount of pages : ";

cin>>pageCount;

}

void showData2(){

cout<<"the amount of pages in this book are as : "<<pageCount<<endl;

}

};

class audio : public book {

public :

float minutes ;

void getData3 ()

{

cout<<"enter the minutes of audio : ";

cin>>minutes;

}

void displayData3()

{

cout<<"the minutes in this audio are : "<<minutes<<endl;

}

};

int main ()

{

book book1;

book1.getData1();

book1.getData2();

book1.display1();

book1.showData2();

audio book2;

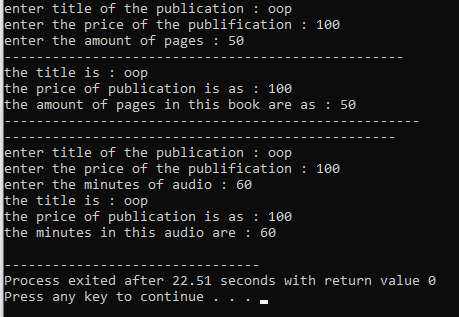
book2.getData1();

book2.getData3();

book2.display1();

book2.displayData3();

return 0;

}

* Program No.2:

// File: Program no.2.cpp

// Date: 22-05-2024

// Name: Abdul Haseeb Arif

// Registration No: 2023-BS-AI-033

//Start with the publicaƟon, book, and tape classes of QuesƟon 1. Add a base class sales that holds an

//array of three floats so that it can record the dollar sales of a parƟcular publicaƟon for the last three

//months. Include a getdata() funcƟon to get three sales amounts from the user, and a putdata() funcƟon

//to display the sales figures. Alter the book and tape classes so they are derived from both publicaƟon

//and sales. An object of class book or tape should input and output sales data along with its other data.

//Write a main() funcƟon to create a book object and a tape object and exercise their input/output

//capabiliƟes.

#include <iostream>

#include <string>

using namespace std;

class Sales {

public:

static const int n = 3;

float salesArr[n];

void getSalesData() {

cout << "Enter the sales for the last three months: " << endl;

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

cin >> salesArr[i];

}

}

void displaySalesData() const {

cout << "Sales data for the last three months: " << endl;

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

cout << "Month " << i + 1 << ": $" << salesArr[i] << endl;

}

}

};

class Company {

public:

string title;

float price;

void getCompanyData() {

cout << "Enter the title of the publication: ";

cin.ignore();

getline(cin, title);

cout << "Enter the price of the publication: ";

cin >> price;

}

void displayCompanyData() const {

cout << "Publication Title: " << title << endl;

cout << "Price: $" << price << endl;

}

};

class Book : public Company, public Sales {

public:

int pageCount;

void getBookData() {

getCompanyData();

cout << "Enter the number of pages in the book: ";

cin >> pageCount;

getSalesData();

}

void displayBookData() const {

displayCompanyData();

cout << "Number of Pages: " << pageCount << endl;

displaySalesData();

}

};

class Audio : public Company, public Sales {

public:

float duration;

void getAudioData() {

getCompanyData();

cout << "Enter the duration of the audio (in minutes): ";

cin >> duration;

getSalesData();

}

void displayAudioData() const {

displayCompanyData();

cout << "Duration (minutes): " << duration << endl;

displaySalesData();

}

};

int main() {

Book book;

cout << "Enter data for the Book:" << endl;

book.getBookData();

cout << "Book Details:" << endl;

book.displayBookData();

Audio audio;

cout << "Enter data for the Audio:" << endl;

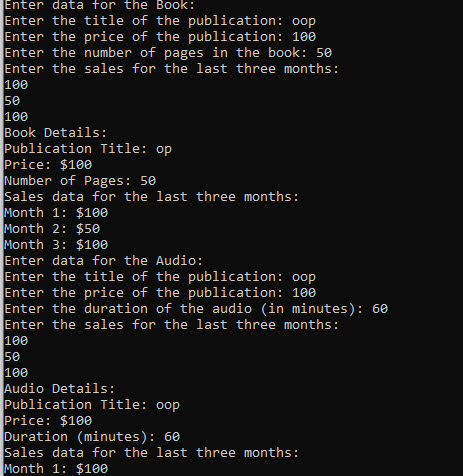
audio.getAudioData();

cout << "Audio Details:" << endl;

audio.displayAudioData();

return 0;

}



* Program No.3:

// File: Program no.3.cpp

// Date: 22-05-2024

// Name: Abdul Haseeb Arif

// Registration No: 2023-BS-AI-033

//Assume that the publisher in QuesƟon 1 and 3 decides to add a third way to distribute books: on computer

//disk, for those who like to do their reading on their laptop. Add a disk class that, like book and tape, is

//derived from publicaƟon. The disk class should incorporate the same member funcƟons as the other

//classes. The data item unique to this class is the disk type: either CD or DVD. You can use an enum type to

//store this item. The user could select the appropriate type by typing c or d.

#include <iostream>

using namespace std;

class publication {

private:

string title;

float price;

public:

void getpublicationdata() {

cout << "Enter Title of book: ";

cin >> title;

cout << "Enter price of book: ";

cin >> price;

}

void putpublicationdata() {

cout << "Title of book is: " << title << endl;

cout << "Price of book is: " << price << endl;

}

};

class book : public publication {

private:

int page\_count;

public:

void getbookdata() {

cout << "Enter Page count of book: ";

cin >> page\_count;

}

void putbookdata() {

cout << "Page count of book is: " << page\_count << endl;

}

};

class tape : public publication {

private:

float minutes;

public:

void gettapedata() {

cout << "Enter minutes of tape: ";

cin >> minutes;

}

void puttapedata() {

cout << "Minutes of tape is: " << minutes << endl;

}

};

enum DiskType { CD, DVD };

class disk : public publication {

private:

DiskType disk\_type;

public:

void getdiskdata() {

char type;

cout << "Enter disk type (c for CD, d for DVD): ";

cin >> type;

if (type == 'c' || type == 'C') {

disk\_type = CD;

} else if (type == 'd' || type == 'D') {

disk\_type = DVD;

} else {

cout << "Invalid disk type. Defaulting to CD." << endl;

disk\_type = CD;

}

}

void putdiskdata() {

cout << "Disk type is: " << (disk\_type == CD ? "CD" : "DVD") << endl;

}

};

int main() {

book book1;

tape tap1;

disk disk1;

book1.getpublicationdata();

book1.getbookdata();

tap1.gettapedata();

disk1.getdiskdata();

book1.putpublicationdata();

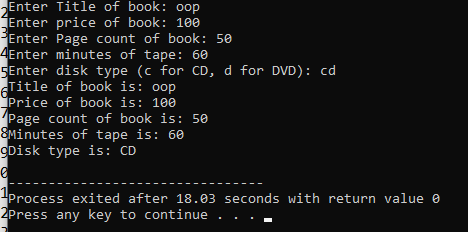
book1.putbookdata();

tap1.puttapedata();

disk1.putdiskdata();

return 0;

}



* Program No.4:

// File: Program no.4.cpp

// Date: 22-05-2024

// Name: Abdul Haseeb Arif

// Registration No: 2023-BS-AI-033

//Derive a class called employee2 from the employee class in the EMPLOY program in this chapter. This new

//class should add a type double data item called compensaƟon, and also an enum type called period to

//indicate whether the employee is paid hourly, weekly, or monthly. For simplicity you can change the

//manager, scienƟst, and laborer classes so they are derived from employee2 instead of employee. However,

//note that in many circumstances it might be more in the spirit of OOP to create a separate base class called

//compensaƟon and three new classes manager2, scienƟst2, and laborer2, and use mulƟple inheritance to

//derive these three classes from the original manager, scienƟst, and laborer classes and from

//compensaƟon. This way none of the original classes needs to be modified

#include <iostream>

#include <string>

using namespace std;

enum Period { HOURLY, WEEKLY, MONTHLY };

class Employee {

protected:

string name;

unsigned long number;

public:

void getdata() {

cout << "Enter name: ";

cin >> name;

cout << "Enter number: ";

cin >> number;

}

void putdata() const {

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

cout << "Number: " << number << "\n";

}

};

class Employee2 : public Employee {

private:

double compensation;

Period period;

public:

void getdata() {

Employee::getdata();

cout << "Enter compensation: ";

cin >> compensation;

int periodInput;

cout << "Enter pay period (0 for Hourly, 1 for Weekly, 2 for Monthly): ";

cin >> periodInput;

period = static\_cast<Period>(periodInput);

}

void putdata() const {

Employee::putdata();

cout << "Compensation: " << compensation << "\n";

cout << "Pay period: ";

switch (period) {

case HOURLY: cout << "Hourly\n"; break;

case WEEKLY: cout << "Weekly\n"; break;

case MONTHLY: cout << "Monthly\n"; break;

}

}

};

class Manager2 : public Employee2 {

private:

string title;

double dues;

public:

void getdata() {

Employee2::getdata();

cout << "Enter title: ";

cin >> title;

cout << "Enter dues: ";

cin >> dues;

}

void putdata() const {

Employee2::putdata();

cout << "Title: " << title << "\n";

cout << "Dues: " << dues << "\n";

}

};

class Scientist2 : public Employee2 {

private:

int publications;

public:

void getdata() {

Employee2::getdata();

cout << "Enter number of publications: ";

cin >> publications;

}

void putdata() const {

Employee2::putdata();

cout << "Publications: " << publications << "\n";

}

};

class Laborer2 : public Employee2 {

// No additional data members

};

int main() {

Manager2 mgr;

Scientist2 sci;

Laborer2 lab;

cout << "Enter manager data:\n";

mgr.getdata();

cout << "\nEnter scientist data:\n";

sci.getdata();

cout << "\nEnter laborer data:\n";

lab.getdata();

cout << "\nManager data:\n";

mgr.putdata();

cout << "\nScientist data:\n";

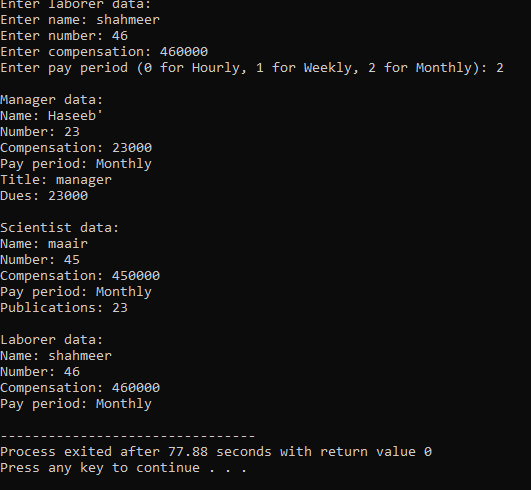
sci.putdata();

cout << "\nLaborer data:\n";

lab.putdata();

return 0;

}



* Program No.5:

// File: Program no.5.cpp

// Date: 22-05-2024

// Name: Abdul Haseeb Arif

// Registration No: 2023-BS-AI-033

//Create a simple inheritance hierarchy for a Shape class, Circle class, and Rectangle class. The Shape class

//should be the base class, and Circle and Rectangle should be derived classes. Implement the following in

//C++:

//Shape Class:

//AƩributes: color (type std::string).

//Methods: A constructor to iniƟalize the color and a method printColor to display the color.

//Circle Class:

//AƩributes: radius (type double).

//Methods: A constructor to iniƟalize the color and radius, a method calculateArea to calculate the area of

//the circle (area = π \* radius \* radius), and a method printArea to display the area.

//Rectangle Class:

//AƩributes: length and width (type double).

//Methods: A constructor to iniƟalize the color, length, and width, a method calculateArea to calculate the

//area of the rectangle (area = length \* width), and a method printArea to display the area.

#include <iostream>

#include <string>

#include <cmath>

using namespace std;

class Shape {

protected:

string color;

public:

Shape(const string& color) : color(color) {}

void printColor() const {

cout << "Color: " << color << endl;

}

};

class Circle : public Shape {

private:

double radius;

public:

Circle(const string& color, double radius) : Shape(color), radius(radius) {}

double calculateArea() const {

return M\_PI \* radius \* radius;

}

void printArea() const {

cout << "Circle Area: " << calculateArea() << endl;

}

};

class Rectangle : public Shape {

private:

double length;

double width;

public:

Rectangle(const string& color, double length, double width) : Shape(color), length(length), width(width) {}

double calculateArea() const {

return length \* width;

}

void printArea() const {

cout << "Rectangle Area: " << calculateArea() << endl;

}

};

int main() {

Circle circle("yellow", 3);

cout << "Circle details:" << endl;

circle.printColor();

circle.printArea();

Rectangle rectangle("red", 2, 4);

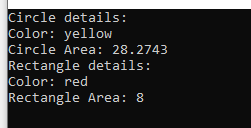
cout << "Rectangle details:" << endl;

rectangle.printColor();

rectangle.printArea();

return 0;

}



* Program No.6:

// File: Program no.6.cpp

// Date: 22-05-2024

// Name: Abdul Haseeb Arif

// Registration No: 2023-BS-AI-033

//Design a class hierarchy for an Employee management system. The base class should be Employee with

//derived classes SalariedEmployee and CommissionEmployee. Each class should have appropriate data

//members and member funcOons to handle the specific a?ributes and behaviors of each type of employee.

//Employee: Should have data members for name, employee ID, and department. It should also have

//member funcOons to get and set these values.

//Salaried Employee: Inherits from Employee and adds a data member for annual Salary. It should have

//member funcOons to get and set the salary, and to calculate the monthly pay.

//Commission Employee: Inherits from Employee and adds data members for sales and commission Rate. It

//should have member funcOons to get and set these values, and to calculate the total pay based on sales

//and commission rate

#include<iostream>

using namespace std;

class Employee {

private:

string name, department;

int EmpID;

public:

void getEmpdata() {

cout << "Enter Employee Name: ";

cin >> name;

cout << "Enter Employee Department: ";

cin >> department;

cout << "Enter Employee ID: ";

cin >> EmpID;

}

void putEmpdata() {

cout << "Employee Name: " << name << endl;

cout << "Employee Department: " << department << endl;

cout << "Employee ID: " << EmpID << endl;

}

};

class SalariedEmployee : public Employee {

private:

int salary;

public:

void getSEmpdata() {

cout << "Enter Employee Salary: ";

cin >> salary;

}

void putSEmpdata() {

cout << "Employee Salary: " << salary << endl;

}

};

class CommissionedEmployee : public Employee {

private:

int salary;

float commissionrate;

public:

void getCEmpdata() {

cout << "Enter Employee Salary: ";

cin >> salary;

cout << "Enter Employee Commission Rate (%): ";

cin >> commissionrate;

}

void putCEmpdata() {

float commission = (salary \* commissionrate) / 100;

float totalSalary = salary + commission;

cout << "Employee Salary after commission: " << totalSalary << endl;

}

};

int main() {

SalariedEmployee haseeb;

CommissionedEmployee maair;

haseeb.getEmpdata();

haseeb.getSEmpdata();

maair.getCEmpdata();

haseeb.putEmpdata();

haseeb.putSEmpdata();

maair.putCEmpdata();

return 0;

}

