OOP

Assignment#3

Submitted By:

Muhammad Usman

4141-FBAS/BSSE/F20

Submitted to:

Mr Muhammad Idrees

Q#3

#include <iostream>

#include <iomanip>

using namespace std;

class pointType

{

public:

pointType()

{

x=0;

y=0;

}

pointType::pointType(double x,double y)

{

this->x = x;

this->y = y;

}

void pointType::setPoint(double x,double y)

{

this->x=x;

this->y=y;

}

void pointType::print()

{

cout<<"("<<x<<","<<y<<")\n";

}

double pointType::getX()

{return x;

}

double pointType::getY()

{return y;

}

private:

double x,y;

};

int main()

{

pointType p2;

double x,y;

cout<<"Enter an x Coordinate for point ";

cin>>x;

cout<<"Enter an y Coordinate for point ";

cin>>y;

p2.setPoint(x,y);

p2.print();

system("pause");

return 0;

}

Q#6

#include <iostream>

#include <cstring> //this takes care of the customer details

//that will be imputed into the program

using namespace std;

//using user-defined struct format to store contact details

struct Contact{

char FirstName [60]; //character string for first name

char LastName [60]; //character string for last name

char PhoneNumber [30]; //character string for phone number

//this takes care of cases where international format is used

char Email [40]; //character string for email

};

int main()

{

//case example for storing values of a contact called John

Contact John;

//strcpy is used to assign character strings

//to user-defined structs

strcpy(John.FirstName,"John");

strcpy(John.LastName, "Doe");

strcpy(John.PhoneNumber,"+1234567890");

strcpy(John.Email, "johndoe @ nomail . com");

//added spaces due to regulations

//this has stored the information of Joe

//printing the contact name

cout<<"The customer is:"<<John.LastName<<" "<<John.FirstName;

//prints out the last name then first name as given

//you can edit it for the desired customers

}

Q#9

#include <iostream>

using std::cout;

using std::cin;

using std::endl;

#include <string>

using std::string;

class Date{

private:

int month;

int day;

int year;

public:

Date();

Date(int month,int day,int year);

void print\_f1(); //format 1

void print\_f2(); //format 2

void print\_f3(); //format 3

};

int main(){

Date myDate;

myDate.print\_f1();

cout<<endl;

myDate.print\_f2();

cout<<endl;

return 0;

}

Date::Date(){

month=1;

day=1;

year=2001;

}

Date::Date(int month,int day,int year){

}

void Date::print\_f1(){

cout<<month<<'/'<<day<<'/'<<year;

}

void Date::print\_f2(){

string Month;

switch(month){//class data member: month

case 1:

Month="January";

break;

case 2:

Month="February";

break;

case 3:

Month="March";

break;

case 4:

Month="April";

break;

case 5:

Month="May";

break;

case 6:

Month="June";

break;

case 7:

Month="July";

break;

case 8:

Month="August";

break;

case 9:

Month="September";

break;

case 10:

Month="October";

break;

case 11:

Month="November";

break;

case 12:

Month="December";

break;

}

cout<<Month<<'/'<<day<<'/'<<year;

}

Q#12

#include<iostream>

#include<string>

using namespace std;

class PersonType

{

public:

void print() const;

void SetName(string first, string last);

string getFirstName() const;

string getLastName() const;

PersonType(string first = "", string last = "");

private:

string FirstName, LastName;

};

void PersonType :: print() const

{

cout << FirstName << " " << LastName << endl;

}

void PersonType :: SetName(string first, string last)

{

FirstName = first;

LastName = last;

cout << "Please Enter Your FIRST Name.\n";

cin >> first;

cout << "Please Enter Your LAST Name.\n";

cin >> last;

}

string PersonType :: getFirstName() const

{

return FirstName;

}

string PersonType :: getLastName() const

{

return LastName;

}

PersonType :: PersonType(string first, string last)

{

FirstName = first;

LastName = last;

}

class DoctorType : public PersonType

{

public:

void SetName(string first, string last);

void SetSpeciality(string Special);

string getSpeciality() const;

string getFirstName() const;

string getLastName() const;

DoctorType(string Special = "");

private:

string FirstName, LastName, Speciality;

};

void DoctorType :: SetSpeciality(string Special)

{

Speciality = Special;

cout << "Please Enter Your SPECIALITY.\n";

cin >> Special;

}

string DoctorType :: getSpeciality() const

{

return Speciality;

}

DoctorType :: DoctorType(string Special = "")

{

Speciality = Special;

}

class DateType

{

public:

void printDate() const;

void SetDate(int Day, int Month, int Year);

int getTheDay() const;

int getTheMonth() const;

int getTheYear() const;

DateType(int Day,int Month, int Year);

private:

int TheDay, TheYear, TheMonth;

};

void DateType :: printDate() const

{

cout << TheDay << " " << TheMonth << " " << TheYear << endl;

}

void DateType :: SetDate(int Day, int Month, int Year)

{

TheDay = Day;

TheMonth = Month;

TheYear = Year;

cout << "Please Enter The Day (DD).\n";

cin >> Day;

cout << "Please Enter The Month (MM).\n";

cin >> Month;

cout << "Please Enter The Year (YYYY).\n";

cin >> Year;

}

int DateType :: getTheDay() const

{

return TheDay;

}

int DateType :: getTheMonth() const

{

return TheMonth;

}

int DateType :: getTheYear() const

{

return TheYear;

}

DateType::DateType(int Day, int Month, int Year)

{

TheDay = Day;

TheMonth = Month;

TheYear = Year;

}

class DateOfBirthType : public DateType

{

public:

void printDOB() const;

void SetDOB(int Day, int Month, int Year);

int getTheDay() const;

int getTheMonth() const;

int getTheYear() const;

private:

int TheDay, TheYear, TheMonth;

};

void DateOfBirthType :: printDOB() const

{

cout << "Patients Date Of Birth: " << TheDay << "/" << TheMonth << "/" << TheYear << endl;

}

void DateOfBirthType :: SetDOB(int Day, int Month, int Year)

{

TheDay = Day;

TheMonth = Month;

TheYear = Year;

cout << "Enter The Patients DAY Of Birth.\n";

cin >> Day;

cout << "Enter The Patients MONTH Of Birth.\n";

cin >> Month;

cout << "Enter The Patients YEAR Of Birth.\n";

cin >> Year;

}

class AdmittanceDateType : public DateType

{

public:

void printAdmittanceDate() const;

void SetAdmittanceDate(int Day, int Month, int Year);

int getTheDay() const;

int getTheMonth() const;

int getTheYear() const;

private:

int TheDay, TheYear, TheMonth;

};

void AdmittanceDateType :: printAdmittanceDate() const

{

cout << "The Patients Admittance Date: " << TheDay << " " << TheMonth << " " << TheYear << endl;

}

void AdmittanceDateType :: SetAdmittanceDate(int Day, int Month, int Year)

{

TheDay = Day;

TheMonth = Month;

TheYear = Year;

cout << "Enter The Patients DAY Of Admittance.\n";

cin >> Day;

cout << "Enter The Patients MONTH Of Admittance.\n";

cin >> Month;

cout << "Enter The Patients YEAR Of Admittance.\n";

cin >> Year;

}

class DischargeDateType : public DateType

{

public:

void printDischargeDate() const;

void SetDischargeDate(int Day, int Month, int Year);

int getTheDay() const;

int getTheMonth() const;

int getTheYear() const;

private:

int TheDay, TheYear, TheMonth;

};

void DischargeDateType :: printDischargeDate() const

{

cout << "The Patients Discharge Date: " << TheDay << " " << TheMonth << " " << TheYear << endl;

}

void DischargeDateType :: SetDischargeDate(int Day, int Month, int Year)

{

TheDay = Day;

TheMonth = Month;

TheYear = Year;

cout << "Enter The Patients DAY Of Discharge.\n";

cin >> Day;

cout << "Enter The Patients MONTH Of Discharge.\n";

cin >> Month;

cout << "Enter The Patients YEAR Of Discharge.\n";

cin >> Year;

}

class PatientType : public PersonType

{

void print() const;

void SetName(string first, string last);

void SetPatientID(int ID);

void SetPatientAge(int Age);

int getPatientAge() const;

int getPatientID() const;

string getFirstName() const;

string getLastName() const;

PatientType(string first = "", string last = "");

PatientType(int ID = 0);

PatientType(int Age = 0);

private:

string FirstName, LastName;

int PatientID, PatientAge;

};

int PatientType::getPatientID() const

{

return PatientID;

}

int PatientType::getPatientAge() const

{

return PatientAge;

}

PatientType::PatientType(string first = "", string last = "")

{

FirstName = first;

LastName = last;

}

PatientType::PatientType(int ID = 0)

{

PatientID = ID;

}

PatientType::PatientType(int Age = 0)

{

PatientID = Age;

}

int main()

{

int choice;

PersonType \*Person = NULL;

cout << "Who would you like to input information for?\n";

cout << " 1 - Doctor\n";

cout << " 2 - Patient\n";

cin >> choice;

if (choice == 1)

{

Person = new DoctorType();

Person->SetName();

Person->SetSpeciality();

}

else if (choice == 2)

{

Person = new PatientType();

Person->;

Person->;

}

return 0;

}

Q#13

class student {

private:

int studentID;

string student\_name;

public:

int getID() // accessor...

{

return studentID;

}

string getname() //accessor...

{

return student\_name;

}

void setvalues(int ID,string name) //mutator..

{

studentID=ID;

student\_name=name;

}

}

Q#14

#ifndef BANKACCOUNT\_H

#define BANKACCOUNT\_H

#include <iostream>

using namespace std;

class bankAccount //set the account number, retrieve the account number, retrieve the balance, deposit and withdraw money, print account information

{

public:

bankAccount(); //sets account number and balance to zero as default;

bankAccount(int, double, double, double);

~bankAccount(){}

void setAccountNum(int); //sets account number

int getAccountNum() const;

double getBalance() const;

void deposit(double); //calculates deposit value

void withdraw(double); //calculates withdraw value

void printAccountInfo() const; //prints info

void setBalance(double);

private:

int accountNum;

//double accountBalance;

protected:

double accountBalance;

};

#endif

//bankAccount.cpp

#include "bankAccount.h"

bankAccount::bankAccount()

{

accountNum = 0;

accountBalance = 0;

};

bankAccount::bankAccount(int n, double b, double d, double w)

{

setAccountNum(n);

setBalance(b);

deposit(d);

withdraw(w);

};

void bankAccount::setAccountNum(int n)

{

accountNum = n;

};

int bankAccount::getAccountNum() const

{

return accountNum;

};

void bankAccount::setBalance(double b)

{

accountBalance = b;

}

double bankAccount::getBalance() const

{

return accountBalance;

};

void bankAccount::deposit(double d)

{

if (d >= 0)

{

accountBalance += d;

}

else

{

cout << "The withdraw ammount is invalid. The withdraw ammount will be set to zero." << endl;

d = 0;

accountBalance += d;

}

};

void bankAccount::withdraw(double w)

{

if (w >= 0)

{

accountBalance -= w;

}

else

{

cout << "The withdraw ammount is invalid. The withdraw ammount will be set to zero." << endl;

w = 0;

accountBalance -= w;

}

};

void bankAccount::printAccountInfo() const

{

cout << "Account Number: " << accountNum << endl;

cout << "Account Balance: " << accountBalance << endl;

};

//checkingAccount.h (derived class)

#ifndef CHECKINGACCOUNT\_H

#define CHECKINGACCOUNT\_H

#include <iostream>

#include "bankAccount.h"

using namespace std;

class checkingAccount : public bankAccount

{

public:

checkingAccount();

~checkingAccount(){}

checkingAccount(int, double, double, double, double, double, double);

void setInterest(double);

void setminBalance(double);

double getminBalance();

void setServiceCharges(double);

double getServiceCharges();

void postInterest(); //verify if the balance is less than the minimum balance, write a check, withdraw (override the method of the base class), and print account information

//void getInterest(double);

void withdraw(double);

void printAccountInfo() const;

//void BalanceVerification();

void CheckAmount(double);

private:

double interest;

double minBalance;

double serviceCharges;

};

#endif

//checkingAccount.cpp (derived class definition)

#include "checkingAccount.h"

checkingAccount::checkingAccount() : bankAccount()

{

interest = 0;

minBalance = 0;

serviceCharges = 0;

}

checkingAccount::checkingAccount(int n, double b, double d, double w, double i, double mb, double sc) : bankAccount(n, b, d, w)

{

setInterest(i);

setminBalance(mb);

setServiceCharges(sc);

}

void checkingAccount::setInterest(double i)

{

interest = i;

}

void checkingAccount::setminBalance(double mb)

{

minBalance = mb;

}

double checkingAccount::getminBalance()

{

return minBalance;

}

void checkingAccount::setServiceCharges(double sc)

{

serviceCharges = sc;

}

double checkingAccount::getServiceCharges()

{

return serviceCharges;

}

void checkingAccount::postInterest()

{

accountBalance \*= (1 + interest / 100);

}

void checkingAccount::CheckAmount(double c)

{

accountBalance -= c;

}

void checkingAccount::printAccountInfo() const

{

cout << "Account Number: " << getAccountNum() << endl;

cout << "Account Balance: " << getBalance() << endl;

cout << "Minimum Balance: " << minBalance << endl;

cout << "Interest: " << interest << endl;

cout << "Service Charge: " << serviceCharges << endl;

//cout << "Post Interest: " <<

};

void checkingAccount::withdraw(double w)

{

accountBalance -= w;

if ((accountBalance) < minBalance)

{

accountBalance -=serviceCharges;

}

}

//main.cpp (testing)

#include "bankAccount.h"

#include "checkingAccount.h"

int main()

{

checkingAccount Customer1;

Customer1.setAccountNum(100584220);

Customer1.setBalance(2500);

Customer1.setServiceCharges(15);

Customer1.setminBalance(350);

Customer1.setInterest(5);

Customer1.deposit(250);

Customer1.withdraw(159.55);

Customer1.CheckAmount(1000);

Customer1.postInterest();

Customer1.printAccountInfo();

system("PAUSE");

return 0;

}