//Author: Linganiso Solethu

//File: dateType.h is the header file

//This file contains specifications for the main program

#ifndef DATETYPE\_H

#define DATETYPE\_H

#include <iostream>

using namespace std;

class dateType{

private:

int day; //variable to store the day

int month; //variable to store the month

int year; //variable to store the year

public:

dateType();

//Default constructor

//The date is set to 0/0/0.

//Postcondition: day = 0; month = 0; year = 0;

dateType(int d,int m,int y);

//Constructor with parameters.

//The date is set according to the parameters.

//Postcondition: day = d; month = m; year = y;

void setDate(int ,int,int );

//Function to set the Date.

//The date is set according to the parameters.

//Postcondition: day = d; month = m; year = y;

// The function checks whether the

// values of day, month and year

// are valid. If a value is invalid, the

// default value 0 is assigned.

void printDate() const;

//This function connot modify the member variables of a variable of type dateType

//Function to output the Date

//Postcondition: The date is printed in the form day/month/year.

};

#endif

//Author: Linganiso Solethu

//File: personType.h is the header file

//This file contains specifications for the main program

#ifndef PERSONTYPE\_H

#define PERSONTYPE\_H

#include <iostream>

using namespace std;

//Base class

class personType{

private:

string firstNam; //variable to store the first name

string lastNam; //variable to store the last name

public:

personType();

//Default constructor

//The set first name and last name as empty string (" ").

//Postcondition: firstNam = " "; lastNam = " ";

personType(string,string);

//Constructor with parameters.

//The first Name and last Name is set according to the parameters.

//Postcondition: firstNam=firstName; lastNam=lastName;

void setFirstName(string); //Function to set first name

string getFirstName() const; //Function to retun fisrt name

void setLastName(string); //Function to set last name

string getLastName() const; //Function to return last name

};

#endif

//Author: Linganiso Solethu

//File: doctorType.h is the header file

//This file contains specifications for the main program

#ifndef DOCTORTYPE\_H

#define DOCTORTYPE\_H

#include <iostream>

using namespace std;

//Sub class of personType which is the base class.

class doctorType: public personType

{

private:

string speciality; //variable to store the speciality of the doctor.

public:

doctorType();

//default constructor

//The speciality is set to " ".

doctorType(string,string,string); //constructor with parameters include the one in the base class.

void setSpeciality(string); //Function to set the speciality.

string getSpeciality() const; //Function to return the speciality.

};

#endif

//Author: Linganiso Solethu

//File: billType.h is the header file

//This file contains specifications for the main program

#ifndef BILLTYPE\_H

#define BILLTYPE\_H

#include <iostream>

using namespace std;

class billType{

private:

string patient\_id; //variable to store patient's ID.

double pharmacyCharges; //Variable to store pharmacy's Charges.

double doctorFee; //Variable to store doctor's fee.

double roomCharges; //Variable to store room's Charges.

public:

billType();

//default constructor.

//The patient\_id is set to " ".

//The pharmacyCharges is set to 0.

//The doctorFee is set to 0.

//The roomCharges is set to 0.

billType(string,double,double,double); //constructor with parameters.

void setPatient\_id(string); //Function to set patient's ID.

string getPatient\_id() const; //Function to return patient's ID.

void setPharmacyCharges(double); //Function to set pharmacy's Charges.

double getPharmacyCharges() const; //Function to return pharmacy's Charges.

void setDoctorFee(double); //Function to set doctor's fee.

double getDoctorFee() const; //Function to return doctor's fee.

void setRoomCharges(double); //Function to set room's Charges.

double getRoomCharges() const; //Function to return room's Charges.

};

#endif

//Author: Linganiso Solethu

//File: patientType.h is the header file

//This file contains specifications for the main program

#ifndef PATIENTTYPE\_H

#define PATIENTTYPE\_H

#include <iostream>

using namespace std;

//Sub class derived from personType which is a base class

class patientType : public personType{

private:

string patient\_id; //Variable to store patient's ID.

int age; //Variable to store age.

string date\_of\_birth; //Variable to store date of birth.

string physicianNam; //Variable to store physician's name.

string admissionDate; //Variable to store admission date.

string dischargedDate; //Variable to store discharged date.

public:

patientType(); //default constructor

patientType(string,string,string,int,string,string,string,string);

//constructor with parameters include the two members from base class.

void setPatient\_id(string); //Function to set patient's ID.

string getPatient\_id() const; //Function return patient's ID.

void setAge(int); //Function to set age.

int getAge() const; //Function to return age.

void setDate\_of\_birth(string); //Function to set date of birth.

string gettDate\_of\_birth() const; //Function to return date of birth.

void setPhysicianNam(string); //Function to set physician's name.

string getPhysicianNam() const; //Function to return physician's name.

void setAdmissionDate(string); //Function to set admission date.

string getAdmissionDate() const; //Function to return admission date.

void setDischargedDate(string); //Function to set discharged date.

string getDischargedDate() const; //Function to set discharged date.

};

#endif

//Author: Linganiso Solethu

//File: Implimentation.h

//Implimentation.h is implementation file

//This file provides the function implimentation or functionality

#include <iostream>

#include "dateType.h"

#include "personType.h"

#include "doctorType.h"

#include "billType.h"

#include "patientType.h"

using namespace std;

dateType ::dateType()

//default constructor that initialize day, month and year to all 0.

{

day=0;

month=0;

year=0;

}

dateType::dateType(int d,int m,int y){

//constructor with parameters

//day = d; month = m; year = y;

//If these values are out of range the member variable d, m and y

//are initialized to 0

if(0<=d&&d<=31)

day=d;

else

day=0;

if(0<=m&&m<=12)

month=m;

else

month=0;

if(y>0)

year=y;

else

year=0;

}

void dateType::setDate(int d,int m,int y){

//The definition of setDate check for the valid value of m,d and y.

//If these values are out of range the member variable d, m and y

//are initialized to 0

if(0<=d&&d<=31)

day=d;

else

day=0;

if(0<=m&&m<=12)

month=m;

else

month=0;

if(y>0)

year=y;

else

year=0;

}

void dateType::printDate() const

{

//The function contain the values that weill call to print them

cout<<day<<"/"<<month<<"/"<<year<<endl; //print the Date

}

personType ::personType(){

//default constructor that initialize both first and last name to empty string.

firstNam=" ";

lastNam=" ";

}

personType ::personType(string firstName,string lastName){

//constructor with parameters which set firstNam=firstName; lastNam=lastName;

firstNam=firstName;

lastNam=lastName;

}

void personType::setFirstName(string firstName){ //Function set firstNam=firstNam;

firstNam=firstNam;

}

string personType::getFirstName() const{ //Function return firstNam;

return firstNam;

}

void personType::setLastName(string lastName){ //Function set lastNam=lastName;

lastNam=lastName;

}

string personType::getLastName() const{ //Function return lastNam;

return lastNam;

}

doctorType::doctorType(){

//default constructor that initialize speciality to empty string.

speciality=" ";

}

doctorType::doctorType(string firstName,string lastName, string spty) :personType(firstName,lastName){

//constructor with parameters. speciality=spty;

//Include first name and the last name from base class personType

speciality=spty;

}

void doctorType::setSpeciality(string spty){

//Function set speciality=spty

speciality=spty;

}

string doctorType::getSpeciality() const{

//Function return speciality

return speciality;

}

billType::billType(){ //default constructor that initialize:

patient\_id=" "; //patient\_id=" ";

pharmacyCharges=0; //pharmacyCharges to 0.

doctorFee=0; //doctorFee to 0.

roomCharges=0; //roomCharges to 0.

}

billType::billType(string id,double Pcharges,double Dfee,double Rcharges) {

//constructor with parameters

patient\_id=id;

pharmacyCharges=Pcharges;

doctorFee=Dfee;

roomCharges=Rcharges;

}

void billType::setPatient\_id(string id){

//Function set patient\_id=id;

patient\_id=id;

}

string billType::getPatient\_id() const{

//Function return patient\_id;

return patient\_id;

}

void billType::setPharmacyCharges(double Pcharges){

//Function set pharmacyCharges=Pcharges;

pharmacyCharges=Pcharges;

}

double billType::getPharmacyCharges() const{

//Function retrieve pharmacyCharges that was set to Pcharges

return pharmacyCharges;

}

void billType::setDoctorFee(double Dfee){

//Function set doctorFee=Dfee;

doctorFee=Dfee;

}

double billType::getDoctorFee() const{

//Function retrieve doctorFee that was set to Dfee;

return doctorFee;

}

void billType::setRoomCharges(double Rcharges){

//Function set Rcharges=Rcharges;

roomCharges=Rcharges;

}

double billType::getRoomCharges() const{

//Function retrieve roomCharges that was set to Rcharges

return roomCharges;

}

patientType::patientType() //default constructor that initialize:

{

patient\_id=" "; //patient\_id to " ".

age=0; //age to 0.

date\_of\_birth=" "; //date of birth to " ".

physicianNam=" "; //Physician's name to " ".

admissionDate=" "; //Admission date to " ".

dischargedDate=" "; //Discharged date to " ".

}

patientType::patientType(string firstName,string lastName,string id,int Age,string dfb,string PNam,string ADate,string Ddate):personType(firstName,lastName)

//constructor with parameters.

//Include first name and the last name from base class personType

{

patient\_id=id;

age=Age;

date\_of\_birth=dfb;

physicianNam=PNam;

admissionDate=ADate;

dischargedDate=Ddate;

}

void patientType::setPatient\_id(string id)

//Function set patient\_id to id;

{

patient\_id=id;

}

string patientType::getPatient\_id() const //This function cannot modify the member // variables of a variable of the type // patientType.

//Function retrieve the patient\_id that was set id.

{

return patient\_id;

}

void patientType::setAge(int Age)

//Function that set age to Age.

{

age=Age;

}

int patientType::getAge() const //This function cannot modify the member // variables of a variable of the type // patientType.

//Function retrieve age that was set to Age.

{

return age;

}

void patientType::setDate\_of\_birth(string dfb)

//Function that set date\_of\_birth to dfb.

{

date\_of\_birth=dfb;

}

string patientType::gettDate\_of\_birth() const //This function cannot modify the member // variables of a variable of the type // patientType.

//Function retrieve date\_of\_birth that was set to dfb.

{

return date\_of\_birth;

}

void patientType::setPhysicianNam(string PNam)

//Functrion that set physicianNam to PNam.

{

physicianNam=PNam;

}

string patientType::getPhysicianNam() const //This function cannot modify the member // variables of a variable of the type // patientType.

//Function retrieve physicianNam that wat set to PNam.

{

return physicianNam;

}

void patientType::setAdmissionDate(string ADate)

//Function set admissionDate to ADate.

{

admissionDate=ADate;

}

string patientType::getAdmissionDate() const //This function cannot modify the member // variables of a variable of the type // patientType.

//Function retrieve that was set to ADate.

{

return admissionDate;

}

void patientType::setDischargedDate(string Ddate){

dischargedDate=Ddate;

}

string patientType::getDischargedDate() const //This function cannot modify the member // variables of a variable of the type // patientType.

//Function retrieve dischargedDate that was set to Ddate.

{

return dischargedDate;

}

//Author: Linganiso Solethu

//publisher : Solethu Linganiso

//Description : Program that computerze the billing

//system of a hospital

//File: mainFunction.h is a main function file

//Date created 06/10/2021

//Due Date:10/10/2021

#include <iostream>

#include "dateType.h"

#include "personType.h"

#include "doctorType.h"

#include "billType.h"

#include "patientType.h"

using namespace std; // use std namespace

// function main begins program execution

int main(){

cout<<endl; //display endl

//display message and endl

cout<<" \*\*\*\*\*\*\*\*\*BILLING SYSTEM OF A HOSPITAL\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<endl;

//display message and endl

cout<<" ================================================"<<endl;

cout<<" Date"<<endl;

cout<<" ================================================"<<endl;

dateType date\_of\_birth(18,8,1998); //initializes its member variables of date of birth // day,month and year to 18, 8 and 1998.

dateType admissionDate(20,5,2020); //initializes its member variables of admission date // day,month and year to 20, 5 and 2020.

dateType dischargedDate(9,10,2021); //initializes its member variables of discharged date // day,month and year to 9, 10 and 2021.

cout<<" Date of birth : "; //Display date of birth of the user

date\_of\_birth.printDate(); //The function output the centents of three member variables of date\_of\_birth.

cout<<" Admission date : "; //Diplay admission date of the user in // hospital

admissionDate.printDate(); //The function output the centents of three // member variables of admissionDate.

cout<<" Discharged date : "; //Display the discharge date of the user in h // hospital

dischargedDate.printDate(); //The function output the centents of three // member variables of dischargedDate.

cout<<endl; //display endl

//display message and endl

cout<<" ================================================"<<endl;

cout<<" Doctor's Info"<<endl;

cout<<" ================================================"<<endl;

doctorType doctor("WILLIAM","SOBHUZA","PLASTIC SURGERY");

cout<<" First Name : "<<doctor.getFirstName()<<endl; //Output the // doctor's first

// name.

cout<<" Last Name : "<<doctor.getLastName()<<endl; //Output the

// doctor's last //name.

cout<<" Specialty : "<<doctor.getSpeciality()<<endl; //Output the // doctor's // speciality

cout<<endl; //display endl

//display message and endl

cout<<" ================================================"<<endl;

cout<<" Patient's Info"<<endl;

cout<<" ================================================"<<endl;

patientType patient("BUZO","MHLALAWEDWA","9806134864087",23," "," "," "," ");

//initializes member variables of First Name, First Name, Patient's ID, Patient's age to BUZ // MHLALAWEDWA, 9806134864087. 23

//Output patient's first name.

cout<<" First Name : "<<patient.getFirstName()<<endl;

//Output patient's last name.

cout<<" Last Name : "<<patient.getLastName()<<endl;

//Output patient's ID.

cout<<" Patient's ID : "<<patient.getPatient\_id()<<endl;

//Output patient's age.

cout<<" Patient's age : "<<patient.getAge()<<" years old"<<endl;

cout<<endl; //display endl

//display message and endl

cout<<" ================================================"<<endl;

cout<<" Billing Details "<<endl;

cout<<" ================================================"<<endl;

billType bill(" ",300,859.45,598.65);

//initializes member variables of billType patient's ID, Pharmacy's Charges,

//Doctor's fee and Room's Charges to " ", 300, 859.45 and 598.65.

//diplay the Pharmacy's Charges.

cout<<" Pharmacy's Charges : R"<<bill.getPharmacyCharges()<<endl;

//display the Doctor's fee.

cout<<" Doctor's fee : R"<<bill.getDoctorFee()<<endl;

//display the Room's Charges.

cout<<" Room's Charges : R"<<bill.getRoomCharges()<<endl;

//Add the charges; display the sum

cout<<" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<" Total charges : R"<<bill.getPharmacyCharges()+

bill.getDoctorFee()+

bill.getRoomCharges()<<endl;

cout<<" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

return 0; // indicate that program ended successfully

}; // end function main