CODE: 1]

#include <iostream>

#include <cmath>

#include <string>

using namespace std;

class Account {

public:

string customerName;

int accountNumber;

float balance;

void initialize(string name, int accNo, float bal) {

customerName = name;

accountNumber = accNo;

balance = bal;

}

void deposit(float amount) {

balance += amount;

cout << "Deposited: " << amount << endl;

}

void displayBalance() {

cout << "Current balance: " << balance << endl;

}

};

class SavAcct : public Account {

public:

void addInterest(float rate, int time) {

float interest = balance \* pow(1 + rate / 100, time) - balance;

balance += interest;

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

displayBalance();

}

void withdraw(float amount) {

if (amount > balance) {

cout << "Insufficient funds!" << endl;

} else {

balance -= amount;

cout << "Withdrew: " << amount << endl;

displayBalance();

}

}

};

class CurAcct : public Account {

const float MIN\_BALANCE = 1000.0;

const float PENALTY = 50.0;

public:

void withdraw(float amount) {

if (amount > balance) {

cout << "Insufficient funds!" << endl;

} else {

balance -= amount;

cout << "Withdrew: " << amount << endl;

checkMinimumBalance();

displayBalance();

}

}

void checkMinimumBalance() {

if (balance < MIN\_BALANCE) {

balance -= PENALTY;

cout << "Penalty Applied: " << PENALTY << endl;

}

}

};

int main() {

SavAcct savAcc;

savAcc.initialize("John Doe", 101, 5000.0);

savAcc.deposit(1000);

savAcc.addInterest(5, 1);

savAcc.withdraw(2000);

CurAcct curAcc;

curAcc.initialize("Jane Smith", 102, 1500.0);

curAcc.deposit(500);

curAcc.withdraw(2000);

curAcc.withdraw(1000);

return 0;

}

CODE:2]

CODE 2:

include<iostream>

using namespace std;

class staff{

public:

int code=0;

string name;

int age;

void display(){

cout<<"Enter code of the employee: "<<endl;

cin>>code;

}

};

class teacher: public staff{

public:

string sub;

int code,age=40 ;

void display(string sub){

cout<<"Please enter the name of teacher whose database you wish to see: "<<endl;

cin>>name;

cout<<"Please enter subject shes supervising for: "<<endl;

cin>>sub;

cout<<name<<endl;

cout<<sub<<endl;

cout<<age<<endl;

}

};

class typist: public staff{

public:

//int sped;

int age=34;

string name;

void emp(){

cout<<"Enter name: "<<endl;

cin>>name;

cout<<age<<endl;

cout<<"Typing speed is 60 words per min."<<endl;

}

};

class officer:public staff{

public:

string post;

void pol(){

cout<<"Enter name of officer: "<<endl;

cin>>name;

cout<<"Age: "<<endl;

cin>>age;

cout<<"Enter posting: "<<endl;

cin>>post;

cout<<name<<endl;

cout<<age<<endl;

cout<<post;

}

};

int main(){

staff s;

s.display();

teacher t;

string sub;

t.display(sub);

typist yp;

yp.emp();

officer of;

of.pol();

// s.display();

return 0;

}

CODE 3]

CODE 3:

#include<iostream>

using namespace std;

class AddAmount{

public:

int amount=50;

//int add;

AddAmount(int add){

cout<<"Enter amount which you wish to deposit: "<<endl;

cin>>add;

cout<<"Updated amount: "<<add+amount<<endl;

}

AddAmount(){

cout<<"Current amount: "<<amount<<endl;

}

};

int main(){

AddAmount a;

int add;

AddAmount b(add);

// b(add);

return 0;

}

CODE 5:

#include<iostream>

using namespace std;

class books{

public:

string author, title, pub;

int price;

void lib(){

int n;

int availc=10;

int reqc;

int a;

int tot=0;

cout<<"Number of boooks you wish to buy: "<<endl;

cin>>n;

for(int i=1;i<=n;i++){

cout<<"Enter book name: "<<endl;

cin>>title;

cout<<"Enter author: "<<endl;

cin>>author;

cout<<"Enter publisher: "<<endl;

cin>>pub;

cout<<"Enter price for books: "<<endl;

cin>>a;

tot+=a;

cout<<tot<<endl;

}

cout<<"Enter the number of copies you require"<<endl;

cin>>reqc;

if(availc >= reqc){

cout<<"Book is available! "<<endl;

//cin>>a;

}

else{

cout<<"Sory this book isnt avalable."<<endl;

}

cout<<reqc \* tot;

}

};

int main(){

books b;

b.lib();

return 0;

}

CODE:3]

CODE 3:

#include<iostream>

using namespace std;

class AddAmount{

public:

int amount=50;

//int add;

AddAmount(int add){

cout<<"Enter amount which you wish to deposit: "<<endl;

cin>>add;

cout<<"Updated amount: "<<add+amount<<endl;

}

AddAmount(){

cout<<"Current amount: "<<amount<<endl;

}

};

int main(){

AddAmount a;

int add;

AddAmount b(add);

// b(add);

return 0;

}

CODE 5:

#include<iostream>

using namespace std;

class books{

public:

string author, title, pub;

int price;

void lib(){

int n;

int availc=10;

int reqc;

int a;

int tot=0;

cout<<"Number of boooks you wish to buy: "<<endl;

cin>>n;

for(int i=1;i<=n;i++){

cout<<"Enter book name: "<<endl;

cin>>title;

cout<<"Enter author: "<<endl;

cin>>author;

cout<<"Enter publisher: "<<endl;

cin>>pub;

cout<<"Enter price for books: "<<endl;

cin>>a;

tot+=a;

cout<<tot<<endl;

}

cout<<"Enter the number of copies you require"<<endl;

cin>>reqc;

if(availc >= reqc){

cout<<"Book is available! "<<endl;

//cin>>a;

}

else{

cout<<"Sory this book isnt avalable."<<endl;

}

cout<<reqc \* tot;

}

};

int main(){

books b;

b.lib();

return 0;

}

CODE:4]

#include <iostream>

#include <string>

using namespace std;

class Person {

public:

string name;

int age;

string address;

void setPersonDetails(string n, int a, string addr) {

name = n;

age = a;

address = addr;

}

void displayPersonDetails() {

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

cout << "Age: " << age << endl;

cout << "Address: " << address << endl;

}

};

class Account : public Person {

public:

int accountNumber;

float balance;

void setAccountDetails(int accNo, float bal) {

accountNumber = accNo;

balance = bal;

}

void displayAccountDetails() {

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

cout << "Balance: " << balance << endl;

}

};

class Admin : public Person {

public:

string adminRole;

void setAdminDetails(string role) {

adminRole = role;

}

void displayAdminDetails() {

cout << "Admin Role: " << adminRole << endl;

}

};

class Network : public Account, public Admin {

public:

void displayNetworkDetails() {

cout << "\n--- Network Details ---\n";

displayPersonDetails();

displayAccountDetails();

displayAdminDetails();

}

};

int main() {

Network netUser;

netUser.setPersonDetails("Alice Johnson", 30, "123 Maple Street");

netUser.setAccountDetails(1001, 5000.75);

netUser.setAdminDetails("Network Administrator");

netUser.displayNetworkDetails();

return 0;

}

CODE:5]

#include <iostream>

#include <cstring>

using namespace std;

class Book {

private:

char \*title, \*author, \*publisher;

float \*price;

int \*stock;

public:

Book() {

title = new char[50];

author = new char[50];

publisher = new char[50];

price = new float;

stock = new int;

}

void inputDetails() {

cout << "\nEnter Book Title: ";

cin.ignore();

cin.getline(title, 50);

cout << "Enter Author: ";

cin.getline(author, 50);

cout << "Enter Publisher: ";

cin.getline(publisher, 50);

cout << "Enter Price: ";

cin >> \*price;

cout << "Enter Stock Quantity: ";

cin >> \*stock;

}

bool searchBook(const char \*searchTitle, const char \*searchAuthor) {

return (strcmp(title, searchTitle) == 0 && strcmp(author, searchAuthor) == 0);

}

void displayDetails() {

cout << "\nTitle: " << title

<< "\nAuthor: " << author

<< "\nPublisher: " << publisher

<< "\nPrice: Rs. " << \*price

<< "\nStock: " << \*stock << endl;

}

void processSale(int quantity) {

if (quantity <= \*stock) {

\*stock -= quantity;

cout << "\nSale Successful! Total Cost: Rs. " << (\*price \* quantity) << endl;

} else {

cout << "\nRequired copies not in stock." << endl;

}

}

~Book() {

delete[] title;

delete[] author;

delete[] publisher;

delete price;

delete stock;

}

};

int main() {

Book \*bookList[10];

int bookCount = 0;

int choice;

do {

cout << "\nBookshop Menu:"

<< "\n1. Add Book"

<< "\n2. Search Book"

<< "\n3. Display All Books"

<< "\n4. Exit"

<< "\nEnter your choice: ";

cin >> choice;

switch (choice) {

case 1:

bookList[bookCount] = new Book;

bookList[bookCount]->inputDetails();

bookCount++;

break;

case 2: {

char searchTitle[50], searchAuthor[50];

cout << "\nEnter Book Title: ";

cin.ignore();

cin.getline(searchTitle, 50);

cout << "Enter Author: ";

cin.getline(searchAuthor, 50);

bool found = false;

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

if (bookList[i]->searchBook(searchTitle, searchAuthor)) {

found = true;

bookList[i]->displayDetails();

int quantity;

cout << "Enter quantity to purchase: ";

cin >> quantity;

bookList[i]->processSale(quantity);

break;

}

}

if (!found) {

cout << "\nBook not found in inventory." << endl;

}

break;

}

case 3:

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

bookList[i]->displayDetails();

}

break;

case 4:

cout << "\nExiting program." << endl;

break;

default:

cout << "\nInvalid choice. Please try again." << endl;

}

} while (choice != 4);

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

delete bookList[i];

}

return 0;

}

CODE:6]

#include <iostream>

using namespace std;

class DB; // Forward declaration of class DB

class DM {

private:

int m;

float cm;

public:

DM() : m(0), cm(0.0) {}

void input() {

cout << "Enter distance in meters and centimeters: ";

cin >> m >> cm;

}

void display() const {

cout << m << " meters and " << cm << " centimeters" << endl;

}

friend DM add(const DM&, const DB&);

};

class DB {

private:

int ft;

float in;

public:

DB() : ft(0), in(0.0) {}

void input() {

cout << "Enter distance in feet and inches: ";

cin >> ft >> in;

}

void display() const {

cout << ft << " feet and " << in << " inches" << endl;

}

friend DM add(const DM&, const DB&);

};

DM add(const DM& dm, const DB& db) {

float totalCm = (dm.m \* 100 + dm.cm) + (db.ft \* 30.48 + db.in \* 2.54);

int meters = totalCm / 100;

float centimeters = totalCm - meters \* 100;

return DM{ meters, centimeters };

}

int main() {

DM dm;

DB db;

dm.input();

db.input();

cout << "\nDistance in meters and centimeters: ";

dm.display();

cout << "\nDistance in feet and inches: ";

db.display();

DM result = add(dm, db);

cout <<"\nResulting distance in meters and centimeters: ";

result.display();

return 0;

}

CODE :7]

CODE 7:

include<iostream>

using namespace std;

class person{

public:

int code;

string name;

void show(){

cout<<"Enter code: "<<endl;

cin>>code;

cout<<"Enter name: "<<endl;

cin>>name;

}

};

class account{

public:

int pay;

void acc(){

cout<<"Enter yearly income: "<<endl;

cin>>pay;

}

};

class admin{

public:

int yr;

void adm(){

cout<<"Enter years of experience: "<<endl;

cin>>yr;

}

};

class Person:public account,public admin{

public:

void info(){

acc();

adm();

}

};

int main(){

Person p;

p.info();

return 0;

}

CODE:8]

#include <iostream>

using namespace std;

class Cinema {

private:

bool seats[3][10];

public:

Cinema() {

for (int i = 0; i < 3; ++i)

for (int j = 0; j < 10; ++j)

seats[i][j] = false;

}

void displaySeats(int showtime) {

cout << "\nSeats for Showtime " << showtime << ":\n";

for (int i = 0; i < 10; ++i) {

cout << "Seat " << i + 1 << ": ";

cout << (seats[showtime][i] ? "Booked" : "Available") << endl;

}

}

void bookSeat(int showtime, int seatNumber) {

if (seatNumber < 1 || seatNumber > 10) {

cout << "Invalid seat number.\n";

return;

}

if (seats[showtime][seatNumber - 1]) {

cout << "Seat " << seatNumber << " is already booked.\n";

} else {

seats[showtime][seatNumber - 1] = true;

cout << "Seat " << seatNumber << " booked successfully.\n";

}

}

void checkAvailability(int showtime) {

int availableSeats = 0;

for (int i = 0; i < 10; ++i) {

if (!seats[showtime][i]) {

availableSeats++;

}

}

cout << "\nAvailable seats for Showtime " << showtime << ": " << availableSeats << endl;

}

};

int main() {

Cinema cinema;

int choice, showtime, seatNumber;

do {

cout << "\nCinema Booking System\n";

cout << "1. Display Seats\n";

cout << "2. Book Seat\n";

cout << "3. Check Availability\n";

cout << "4. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter showtime (0 for 1 PM, 1 for 5 PM, 2 for 8:30 PM): ";

cin >> showtime;

cinema.displaySeats(showtime);

break;

case 2:

cout << "Enter showtime (0 for 1 PM, 1 for 5 PM, 2 for 8:30 PM): ";

cin >> showtime;

cout << "Enter seat number (1 to 10): ";

cin >> seatNumber;

cinema.bookSeat(showtime, seatNumber);

break;

case 3:

cout << "Enter showtime (0 for 1 PM, 1 for 5 PM, 2 for 8:30 PM): ";

cin >> showtime;

cinema.checkAvailability(showtime);

break;

case 4:

cout << "Exiting the system.\n";

break;

default:

cout << "Invalid choice. Please try again.\n";

}

} while (choice != 4);

return 0;

}

CODE:9]

#include <iostream>

#include <string>

using namespace std;

class Book {

private:

int classMark;

string title;

bool onLoan;

bool reserved;

public:

Book(int cm, string t)

: classMark(cm), title(t), onLoan(false), reserved(false) {}

void checkOut() {

if (onLoan) {

cout << "Book is already on loan.\n";

} else {

onLoan = true;

cout << "Book checked out successfully.\n";

}

}

void reserve() {

if (reserved) {

cout << "Book is already reserved.\n";

} else {

reserved = true;

cout << "Book reserved successfully.\n";

}

}

void inquire() const {

cout << "Class Mark: " << classMark << "\n";

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

cout << "Status: ";

if (onLoan) {

cout << "On loan";

} else if (reserved) {

cout << "Reserved";

} else {

cout << "Available";

}

cout << "\n";

}

void returnBook() {

if (onLoan) {

onLoan = false;

cout << "Book returned successfully.\n";

} else {

cout << "This book was not on loan.\n";

}

}

bool isOnLoan() const { return onLoan; }

bool isReserved() const { return reserved; }

};

class Library {

private:

Book\* books[100];

int bookCount;

public:

Library() : bookCount(0) {}

void addBook(int cm, string t) {

books[bookCount++] = new Book(cm, t);

cout << "Book added successfully.\n";

}

void checkOutBook(int cm) {

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

if (books[i]->getClassMark() == cm) {

books[i]->checkOut();

return;

}

}

cout << "Book not found.\n";

}

void reserveBook(int cm) {

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

if (books[i]->getClassMark() == cm) {

books[i]->reserve();

return;

}

}

cout << "Book not found.\n";

}

void inquireBook(int cm) const {

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

if (books[i]->getClassMark() == cm) {

books[i]->inquire();

return;

}

}

cout << "Book not found.\n";

}

void returnBook(int cm) {

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

if (books[i]->getClassMark() == cm) {

books[i]->returnBook();

return;

}

}

cout << "Book not found.\n";

}

void libraryStatus() const {

int available = 0, onLoan = 0, reserved = 0;

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

if (books[i]->isOnLoan()) {

++onLoan;

} else if (books[i]->isReserved()) {

++reserved;

} else {

++available;

}

}

cout << "\nLibrary Status:\n";

cout << "Total books: " << bookCount << "\n";

cout << "Available: " << available << "\n";

cout << "On loan: " << onLoan << "\n";

cout << "Reserved: " << reserved << "\n";

}

};

int main() {

Library lib;

int choice, cm;

string title;

do {

cout << "\nLibrary Management System\n";

cout << "1. Add Book\n";

cout << "2. Check Out Book\n";

cout << "3. Reserve Book\n";

cout << "4. Inquire Book\n";

cout << "5. Return Book\n";

cout << "6. Library Status\n";

cout << "7. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter class mark: ";

cin >> cm;

cin.ignore();

cout << "Enter title: ";

getline(cin, title);

lib.addBook(cm, title);

break;

case 2:

cout << "Enter class mark: ";

cin >> cm;

lib.checkOutBook(cm);

break;

case 3:

cout << "Enter class mark: ";

cin >> cm;

lib.reserveBook(cm);

break;

case 4:

cout << "Enter class mark: ";

cin >> cm;

lib.inquireBook(cm);

break;

case 5:

cout << "Enter class mark: ";

cin >> cm;

lib.returnBook(cm);

break;

case 6:

lib.libraryStatus();

break;

case 7:

cout << "Exiting...\n";

break;

default:

cout << "Invalid choice. Try again.\n";

}

} while (choice != 7);

return 0;

}

CODE:10]

#include <iostream>

using namespace std;

class Employee {

public:

int age;

int yearsOfService;

int salary;

void setDetails(int a, int y, int s) {

age = a;

yearsOfService = y;

salary = s;

}

void display() {

cout << "Age: " << age << endl;

cout << "Years of Service: " << yearsOfService << endl;

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

}

};

int main() {

Employee emp1, emp2;

emp1.setDetails(30, 5, 50000);

emp2.setDetails(40, 10, 70000);

cout << "Employee 1 Details:" << endl;

emp1.display();

cout << "\nEmployee 2 Details:" << endl;

emp2.display();

return 0;

}

Code:11

#include <iostream>

using namespace std;

class Mammals {

public:

void mammalInfo() {

cout << "I am mammal" << endl;

}

};

class MarineAnimals {

public:

void marineInfo() {

cout << "I am a marine animal" << endl;

}

};

class BlueWhale : public Mammals, public MarineAnimals {

public:

void blueWhaleInfo() {

cout << "I belong to both the categories: Mammals as well as Marine Animals" << endl;

}

};

int main() {

Mammals mammal;

MarineAnimals marine;

BlueWhale bluewhale;

mammal.mammalInfo();

marine.marineInfo();

bluewhale.blueWhaleInfo();

bluewhale.mammalInfo();

bluewhale.marineInfo();

return 0;

}

Code 12

#include <iostream>

using namespace std;

class Fruit {

protected:

int totalFruits;

public:

Fruit() {

totalFruits = 0;

}

void addFruits(int count) {

totalFruits += count;

}

int getTotalFruits() {

return totalFruits;

}

};

class Apples : public Fruit {

private:

int apples;

public:

Apples(int count) {

apples = count;

addFruits(count);

}

void printApples() {

cout << "Number of apples: " << apples << endl;

}

};

class Mangoes : public Fruit {

private:

int mangoes;

public:

Mangoes(int count) {

mangoes = count;

addFruits(count);

}

void printMangoes() {

cout << "Number of mangoes: " << mangoes << endl;

}

};

int main() {

Apples appleBasket(5);

Mangoes mangoBasket(7);

appleBasket.printApples();

mangoBasket.printMangoes();

int total = appleBasket.getTotalFruits() + mangoBasket.getTotalFruits();

cout << "Total number of fruits: " << total << endl;

return 0;

}

Code 13

#include <iostream>

#include <vector>

using namespace std;

int rollCounter = 1;

class Marks {

protected:

int rollNo;

string name;

int physicsMarks;

int chemistryMarks;

int mathMarks;

public:

Marks(string n) {

name = n;

rollNo = rollCounter++;

physicsMarks = 0;

chemistryMarks = 0;

mathMarks = 0;

}

void setPhysicsMarks(int m) {

physicsMarks = m;

}

void setChemistryMarks(int m) {

chemistryMarks = m;

}

void setMathMarks(int m) {

mathMarks = m;

}

int getTotalMarks() {

return physicsMarks + chemistryMarks + mathMarks;

}

void display() {

cout << "Roll No: " << rollNo << ", Name: " << name << endl;

cout << "Physics: " << physicsMarks << ", Chemistry: " << chemistryMarks << ", Mathematics: " << mathMarks << endl;

cout << "Total Marks: " << getTotalMarks() << endl;

}

int getPhysicsMarks() { return physicsMarks; }

int getChemistryMarks() { return chemistryMarks; }

int getMathMarks() { return mathMarks; }

};

int main() {

int n;

cout << "Enter number of students: ";

cin >> n;

vector<Marks> students;

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

string name;

int phy, chem, math;

cout << "Enter name of student " << i + 1 << ": ";

cin >> name;

Marks student(name);

cout << "Enter Physics marks: ";

cin >> phy;

student.setPhysicsMarks(phy);

cout << "Enter Chemistry marks: ";

cin >> chem;

student.setChemistryMarks(chem);

cout << "Enter Mathematics marks: ";

cin >> math;

student.setMathMarks(math);

students.push\_back(student);

}

int totalPhysics = 0, totalChemistry = 0, totalMath = 0;

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

students[i].display();

totalPhysics += students[i].getPhysicsMarks();

totalChemistry += students[i].getChemistryMarks();

totalMath += students[i].getMathMarks();

cout << endl;

}

float avgPhysics = totalPhysics / float(n);

float avgChemistry = totalChemistry / float(n);

float avgMath = totalMath / float(n);

cout << "Class Average in Physics: " << avgPhysics << endl;

cout << "Class Average in Chemistry: " << avgChemistry << endl;

cout << "Class Average in Mathematics: " << avgMath << endl;

return 0;

}

Code 14

#include <iostream>

using namespace std;

class Vehicle {

protected:

float mileage;

float price;

public:

void setVehicleData(float m, float p) {

mileage = m;

price = p;

}

void displayVehicleData() {

cout << "Mileage: " << mileage << " km/l" << endl;

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

}

};

class Car : public Vehicle {

protected:

float ownershipCost;

int warranty;

int seatingCapacity;

string fuelType;

public:

void setCarData(float oc, int w, int sc, string ft) {

ownershipCost = oc;

warranty = w;

seatingCapacity = sc;

fuelType = ft;

}

void displayCarData() {

cout << "Ownership Cost: $" << ownershipCost << endl;

cout << "Warranty: " << warranty << " years" << endl;

cout << "Seating Capacity: " << seatingCapacity << endl;

cout << "Fuel Type: " << fuelType << endl;

}

};

class Bike : public Vehicle {

protected:

int cylinders;

int gears;

string coolingType;

string wheelType;

float fuelTankSize;

public:

void setBikeData(int cyl, int g, string ct, string wt, float fts) {

cylinders = cyl;

gears = g;

coolingType

code 15

#include <iostream>

using namespace std;

class Shape {

public:

void display() {

cout << "This is a shape" << endl;

}

};

class Polygon : public Shape {

public:

void display() {

cout << "Polygon is a shape" << endl;

}

};

class Rectangle : public Polygon {

public:

void display() {

cout << "Rectangle is a polygon" << endl;

}

};

class Triangle : public Polygon {

public:

void display() {

cout << "Triangle is a polygon" << endl;

}

};

class Square : public Rectangle {

public:

void display() {

cout << "Square is a rectangle" << endl;

}

};

int main() {

Shape s;

Polygon p;

Rectangle r;

Triangle t;

Square sq;

s.display();

p.display();

r.display();

t.display();

sq.display();

return 0;

}

Code 16

#include <iostream>

using namespace std;

class RBI {

public:

virtual void interestRate() {

cout << "Minimum interest rate is 4%" << endl;

}

virtual void minBalance() {

cout << "Minimum balance should be Rs. 1000" << endl;

}

virtual void maxWithdrawal() {

cout << "Maximum withdrawal limit is Rs. 25000" << endl;

}

};

class SBI : public RBI {

public:

void interestRate() {

cout << "SBI interest rate is 4.5%" << endl;

}

};

class ICICI : public RBI {

public:

void interestRate() {

cout << "ICICI interest rate is 5%" << endl;

}

};

class HDFC : public RBI {

public:

void interestRate() {

cout << "HDFC interest rate is 5.5%" << endl;

}

};

int main() {

RBI\* bank;

SBI sbi;

ICICI icici;

HDFC hdfc;

bank = &sbi;

bank->interestRate();

bank->minBalance();

bank->maxWithdrawal();

bank = &icici;

bank->interestRate();

bank->minBalance();

bank->maxWithdrawal();

bank = &hdfc;

bank->interestRate();

bank->minBalance();

bank->maxWithdrawal();

return 0;

}

Code17

#include <iostream>

using namespace std;

class RBI {

protected:

float interestRate;

float minBalance;

float maxWithdrawal;

public:

RBI() {

interestRate = 4.0;

minBalance = 1000;

maxWithdrawal = 25000;

}

virtual void showBankPolicies() {

cout << "RBI Guidelines:" << endl;

cout << "Interest Rate: " << interestRate << "%" << endl;

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

cout << "Max Withdrawal: Rs. " << maxWithdrawal << endl;

}

};

class SBI : public RBI {

public:

void showBankPolicies() {

interestRate = 4.5;

cout << "SBI Policies:" << endl;

cout << "Interest Rate: " << interestRate << "%" << endl;

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

cout << "Max Withdrawal: Rs. " << maxWithdrawal << endl;

}

};

class ICICI : public RBI {

public:

void showBankPolicies() {

interestRate = 5.0;

cout << "ICICI Policies:" << endl;

cout << "Interest Rate: " << interestRate << "%" << endl;

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

cout << "Max Withdrawal: Rs. " << maxWithdrawal << endl;

}

};

class Customer {

public:

string name;

int age;

Customer(string n, int a) {

name = n;

age = a;

}

void displayCustomer() {

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

cout << "Age: " << age << endl;

}

};

class Account {

public:

int accNumber;

float balance;

Account(int accNo, float bal) {

accNumber = accNo;

balance = bal;

}

void displayAccount() {

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

cout << "Balance: Rs. " << balance << endl;

}

};

int main() {

Customer c1("Ravi", 30);

Account a1(123456, 20000);

SBI sbi;

ICICI icici;

c1.displayCustomer();

a1.displayAccount();

cout << endl;

sbi.showBankPolicies();

cout << endl;

icici.showBankPolicies();

return 0;

}

Code18

#include <iostream>

using namespace std;

class Student {

private:

string name;

public:

Student() {

name = "Unknown";

}

Student(string n) {

name = n;

}

void displayName() {

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

}

};

int main() {

Student s1;

Student s2("Amit");

s1.displayName();

s2.displayName();

return 0;

}

Code 19

#include <iostream>

using namespace std;

class Rectangle {

private:

int length;

int breadth;

public:

Rectangle() {

length = 0;

breadth = 0;

}

Rectangle(int l, int b) {

length = l;

breadth = b;

}

Rectangle(int val) {

length = val;

breadth = val;

}

int area() {

return length \* breadth;

}

};

int main() {

Rectangle r1;

Rectangle r2(4, 5);

Rectangle r3(6);

cout << "Area of rectangle with no parameters: " << r1.area() << endl;

cout << "Area of rectangle with two parameters (4,5): " << r2.area() << endl;

cout << "Area of rectangle with one parameter (6): " << r3.area() << endl;

return 0;

}