1.

package assignment;

public class AccountSettings {

public long accNum;

public double balance;

public Person accHolder;

public void deposit(double bal) {

balance+=bal;

}

public void withdraw(double bal){

double res = balance-bal;

if(res <= 500) {

System.out.println("Minimum balance cannot be WithDraw");

return;

}

balance-=bal;

}

public double getBalance() {

return balance;

}

public long getAccNum() {

return accNum;

}

public void setAccNum(long accNum) {

this.accNum = accNum;

}

public Person getAccHolder() {

return accHolder;

}

public void setAccHolder(Person accHolder) {

this.accHolder = accHolder;

}

public void setBalance(double balance) {

this.balance = balance;

}

public static void main(String[] args) {

Person Per = new Person();

Per.setName("Vikas");

Per.setBalance(5000);

Person Pe = new Person();

Pe.setName("Akash");

Pe.setBalance(4000);

Per.deposit(2000);

Pe.withdraw(2000);

SavingsAccount savings = new SavingsAccount();

savings.withdraw(2000);

System.out.println(" account balance : "+Per.balance);

System.out.println("Pe balance: "+Pe.balance);

}

}

class Person extends AccountSettings{

String Name;

float age;

public String getName() {

return Name;

}

public void setName(String name) {

Name = name;

}

public float getAge() {

return age;

}

public void setAge(float age) {

this.age = age;

}

}

class SavingsAccount extends AccountSettings{

final int minimumBalance = 500;

@Override

public void withdraw(double bal) {

double res = balance - bal;

if(res <= 500) {

System.out.println("Cannot WithDraw with minimum balance");

return;

}

balance-=bal;

}

}

class CurrentAccount extends AccountSettings{

double overdraftLimit = 500;

@Override

public void withdraw(double bal) {

double res = balance - bal;

if(res <= 500) {

System.out.println("true");

}

}

}

2.

package.com.assignment;

abstract class Item {

private int id;

private String title;

private int numCopies;

public Item(int id, String title, int numCopies) {

this.id = id;

this.title = title;

this.numCopies = numCopies;

}

public int getId() {

return id;

}

public String getTitle() {

return title;

}

public int getNumCopies() {

return numCopies;

}

public void checkIn() {

numCopies++;

}

public void checkOut() {

if (numCopies > 0) {

numCopies--;

}

}

public void addItem(int n) {

numCopies += n;

}

@Override

public String toString() {

return "ID: " + id + ", Title: " + title + ", Copies: " + numCopies;

}

public void print() {

System.out.println(this.toString());

}

}

abstract class WrittenItem extends Item {

private String author;

public WrittenItem(int id, String title, int numCopies, String author) {

super(id, title, numCopies);

this.author = author;

}

public String getAuthor() {

return author;

}

@Override

public String toString() {

return super.toString() + ", Author: " + author;

}

}

class Book extends WrittenItem {

public Book(int id, String title, int numCopies, String author) {

super(id, title, numCopies, author);

}

@Override

public void print() {

System.out.println("Book Info: " + this.toString());

}

}

class JournalPaper extends WrittenItem {

private int yearPublished;

public JournalPaper(int id, String title, int numCopies, String author, int yearPublished) {

super(id, title, numCopies, author);

this.yearPublished = yearPublished;

}

public int getYearPublished() {

return yearPublished;

}

@Override

public String toString() {

return super.toString() + ", Year Published: " + yearPublished;

}

@Override

public void print() {

System.out.println("Journal Paper Info: " + this.toString());

}

}

abstract class MediaItem extends Item {

private int runtime;

public MediaItem(int id, String title, int numCopies, int runtime) {

super(id, title, numCopies);

this.runtime = runtime;

}

public int getRuntime() {

return runtime;

}

@Override

public String toString() {

return super.toString() + ", Runtime: " + runtime + " minutes";

}

}

class Video extends MediaItem {

private String director;

private String genre;

private int yearReleased;

public Video(int id, String title, int numCopies, int runtime, String director, String genre, int yearReleased) {

super(id, title, numCopies, runtime);

this.director = director;

this.genre = genre;

this.yearReleased = yearReleased;

}

public String getDirector() {

return director;

}

public String getGenre() {

return genre;

}

public int getYearReleased() {

return yearReleased;

}

@Override

public String toString() {

return super.toString() + ", Director: " + director + ", Genre: " + genre + ", Year Released: " + yearReleased;

}

@Override

public void print() {

System.out.println("Video Info: " + this.toString());

}

}

class CD extends MediaItem {

private String artist;

private String genre;

public CD(int id, String title, int numCopies, int runtime, String artist, String genre) {

super(id, title, numCopies, runtime);

this.artist = artist;

this.genre = genre;

}

public String getArtist() {

return artist;

}

public String getGenre() {

return genre;

}

@Override

public String toString() {

return super.toString() + ", Artist: " + artist + ", Genre: " + genre;

}

@Override

public void print() {

System.out.println("CD Info: " + this.toString());

}

}

public class LibraryClient {

public static void main(String[] args) {

Book book1 = new Book(1, "The Great Gatsby", 3, "F. Scott Fitzgerald");

JournalPaper jp1 = new JournalPaper(2, "Quantum Computing", 5, "Albert Einstein", 1920);

Video video1 = new Video(3, "Inception", 2, 148, "Christopher Nolan", "Sci-Fi", 2010);

CD cd1 = new CD(4, "Abbey Road", 4, 47, "The Beatles", "Rock");

book1.print();

jp1.print();

video1.print();

cd1.print();

}

}