**TEST-7**

1. Design a class BankAccount with properties accountNumber and balance, and methodsdeposit() and withdraw(). Extend this class with subclasses SavingsAccount andCheckingAccount. Implement specific rules such as minimum balance requirements andinterest calculation for savings accounts.

**PROGRAM**

public class test1

{

        static class Bankaccount

        {

            public void drive()

            {

                int amount=1000;

                System.out.println("withdraw."+amount);

            }

        }

        static class deposite extends Bankaccount

        {

            public void drive()

            {

                super.drive();

                int amount1=1000;

                int amount2=100000;

                System.out.println("deposite..."+amount2);

                int result=0;

                result=amount2-amount1;

                System.out.println("remaining amount"+result);

            }

        }

        public static void main(String[] args)

        {

            deposite cr=new deposite();

            cr.drive();

        }

}

1. Create a base class GameCharacter with properties name, health, and level. Extend thisclass with subclasses Warrior, Mage, and Archer. Implement methods such as attack()and defend() differently for each subclass, showcasing polymorphism through inheritance.

**PROGRAM**

public class test2 {

    public static void main(String[] args)

    {

         GameCharacter game=new GameCharacter();

         GameCharacter war=new warrior();

         GameCharacter mag=new Mage();

         GameCharacter arc=new Archer();

         arc.attack();

         arc.defend();

         war.attack();

         war.defend();

         mag.attack();

         mag.defend();

    }

}

class GameCharacter{

    String name;

    int health,level;

    public void attack()

    {

          System.err.println("They Are Coming To Attack Our Teritory");

    }

    public void defend()

    {

       System.err.println("Get Ready To Defend");

    }

}

class warrior extends GameCharacter

{

    @Override

    public void attack()

    {

          System.err.println("Get Ready Warrior's They Are Coming To Attack Our Teritory");

    }

    @Override

    public void defend()

    {

       System.err.println("Get Ready To Defend");

    }

}

class Mage extends GameCharacter

{

    @Override

    public void attack()

    {

          System.err.println("Get Ready Mage They Are Coming To Attack Our Teritory");

    }

    @Override

    public void defend()

    {

       System.err.println("Get Ready To Defend");

    }

}

class Archer extends GameCharacter

{

    @Override

    public void attack()

    {

          System.err.println("Get Ready Archer They Are Coming To Attack Our Teritory");

    }

    @Override

    public void defend()

    {

       System.err.println("Get Ready To Defend");

    }

}

1. Design a class Product with properties productId, name, and price. Extend this class withsubclasses Electronics and Clothing. Implement methods to calculate discounts based onmembership status for electronics and seasonal sales for clothing.

**PROGRAM**

import java.util.Scanner;

public class test3 {

    static Scanner scanner=new Scanner(System.in);

    public static void main(String[] args)

    {

        Electronics ect=new Electronics();

        Clothing clt=new Clothing();

        ect.discounts();

    }

}

class properties{

    static Scanner scanner=new Scanner(System.in);

    String name,mship,seasonalsale;

    int id;

    float price;

}

class Electronics extends properties

{

   public void discounts()

   {

    int disc=0;

    System.out.print("Enter The Member Ship Type:-");

    mship=scanner.nextLine();

    if (mship=="gold")

    {

       disc=10;

       System.out.print("Enter The Price:-");

       int price=scanner.nextInt();

       scanner.nextLine();

       int finall=(price\*disc)/100;

       System.out.println("The Final Price After Discount Is:-"+(price-finall));

    }

    else if (mship=="silver")

    {

       disc=5;

       System.out.print("Enter The Price:-");

       int price=scanner.nextInt();

       scanner.nextLine();

       int finall=(price\*disc)/100;

       System.out.println("The Final Price After Discount Is:-"+finall);

    }

    else

    {

       disc=3;

       System.out.print("Enter The Price:-");

       int price=scanner.nextInt();

       scanner.nextLine();

       int finall=(price\*disc)/100;

       System.out.println("The Final Price After Discount Is:-"+finall);

    }

   }

}

class Clothing extends properties

{

    int price,disc=0,finall;

    public void discountt(Boolean result)

    {

       if (result.equals("true"))

       {

        disc=10;

        System.out.print("Enter The Price:-");

        int finall=(price\*disc)/100;

        System.out.println("The Final Price After Discount Is:-"+finall);

       }

       else{

        System.out.print("Enter The Price:-");

        System.out.println("The Final Price After Discount Is:-"+finall);

       }

    }

}

1. Design a class LibraryItem with properties title, author, and year. Extend this class withsubclasses Book and DVD. Implement methods for checking in and out items, anddisplay detailed information for each item type.

**PROGRAM**

from datetime import datetime

class LibraryItem:

    def \_\_init\_\_(self, title, author, year):

        self.title = title

        self.author = author

        self.year = year

        self.checked\_out = False

        self.checkout\_date = None

    def check\_out(self):

        if not self.checked\_out:

            self.checked\_out = True

            self.checkout\_date = datetime.now()

            print(f"{self.title} checked out on {self.checkout\_date.strftime('%Y-%m-%d')}")

        else:

            print(f"{self.title} is already checked out")

    def check\_in(self):

        if self.checked\_out:

            self.checked\_out = False

            self.checkout\_date = None

            print(f"{self.title} checked in")

        else:

            print(f"{self.title} is not checked out")

    def display\_info(self):

        print(f"Title: {self.title}")

        print(f"Author: {self.author}")

        print(f"Year: {self.year}")

        if self.checked\_out:

            print(f"Checked out on: {self.checkout\_date.strftime('%Y-%m-%d')}")

        else:

            print("Available")

class Book(LibraryItem):

    def \_\_init\_\_(self, title, author, year, pages):

        super().\_\_init\_\_(title, author, year)

        self.pages = pages

    def display\_info(self):

        super().display\_info()

        print(f"Pages: {self.pages}")

class DVD(LibraryItem):

    def \_\_init\_\_(self, title, author, year, runtime):

        super().\_\_init\_\_(title, author, year)

        self.runtime = runtime

    def display\_info(self):

        super().display\_info()

        print(f"Runtime: {self.runtime} minutes")

book = Book("To Kill a Mockingbird", "Harper Lee", 1960, 281)

book.display\_info()

book.check\_out()

book.display\_info()

book.check\_in()

book.display\_info()

dvd = DVD("The Shawshank Redemption", "Frank Darabont", 1994, 142)

dvd.display\_info()

dvd.check\_out()

dvd.display\_info()

dvd.check\_in()

dvd.display\_info()