Java OOPS

1. Create a class called "Person" with attributes "name" and "age". Also create a constructor and getter methods for the attributes.

Program:

public class Person {

private String name;

private int age;

public Person(String name, int age) {

this.name = name;

this.age = age;

}

public String getName() {

return this.name;

}

public int getAge() {

return this.age;

}

public static void main(String[] args) {

Person p1 = new Person("Vinoth", 26);

System.***out***.println("Name: " + p1.getName());

System.***out***.println("Age: " + p1.getAge());

}

}

Output:

Name: Vinoth

Age: 26

2. From the below image, Create an implementation for a java class named Employee that represents an employee with attributes including ID, name, and salary. The class includes a method called raise Salary(percent), which updates the salary by a specified percentage

Employee

-id:int -firstName: String

-lastName: String

-salary: int

+Employee (id: int, firstName: String,

lastName: String, salary: int)

+getID(): int

+getFirstName(): String

+getLastName(): String

+getName(): String

+getSalary(): int

+setSalary (salary: int):void

+getAnnualSalary(): int

+raiseSalary (int percent): int +toString(): String

"firstName Lastname"

salary \* 12

Increase the salary by the percent and

return the new salary

"Employee [id=?, name=firstName Lastname, salary=?]"

Program:

public class Employee {

private int id;

private String firstname;

private String lastname;

private int salary;

public Employee(int id, String firstname, String lastname, int salary) {

this.id=id;

this.firstname=firstname;

this.lastname=lastname;

this.salary=salary;

}

public int getId() {

return this.id;

}

public String getFirstname() {

return this.firstname;

}

public String getLastname() {

return this.lastname;

}

public String getName() {

return this.firstname +" "+ this.lastname;

}

public int getSalary() {

return this.salary;

}

public void setSalary() {

return;

}

public int getAnnualsalary() {

return this.salary\*12;

}

public int raiseSalary(int percent) {

this.salary=this.salary + (this.salary\*percent / 100);

return this.salary;

}

public String toString() {

return "Employee [id=" + id + ", name=" + getName() + ", salary=" + salary + "]";

}

public static void main(String[] args) {

Employee e1= new Employee(1234, "Vinoth", "Kumar", 30000);

System.***out***.println(e1.toString());

e1.raiseSalary(10);

System.***out***.println("After salary raise " +e1.toString());

}

}

Output:

Employee [id=1234, name=Vinoth Kumar, salary=30000]

After salary raise Employee [id=1234, name=Vinoth Kumar, salary=33000]

3. Create a class circle class with radius as data member. Create two constructors (no argument, and two arguments) and a method to calculate Circumference.

public class Circle {

private double radius;

public Circle() {

this.radius=0.0;

}

public Circle(double radius) {

this.radius=radius;

}

public double calcCircumference() {

return 2\*Math.***PI***\*this.radius;

}

public static void main(String[] args) {

Circle c1= new Circle(10.0);

System.***out***.println("Circumference of the circle: "+c1.calcCircumference());

}

}

Output:

Circumference of the circle: 62.83185307179586

4. Create a class Account class with balance as data member. Create two constructors (no argument, and two arguments) and methods to withdraw and deposit balance.

Program:

public class Account {

private double balance;

public Account() {

this.balance=0.0;

}

public Account(double balance) {

this.balance= balance;

}

public void deposit(double amount) {

this.balance += amount;

}

public void withdrawl(double amount) {

if(amount <= this.balance) {

this.balance -= amount;

}else {

System.***out***.println("Insufficient balance");

}

}

public double getbalance() {

return this.balance;

}

public static void main(String[] args) {

Account a1= new Account(10000.0);

a1.deposit(500.0);

System.***out***.println("Balance after deposit: "+a1.getbalance());

a1.withdrawl(2000.0);

System.***out***.println("Balance after withdrawal: "+a1.getbalance());

}

}

Output:

Balance after deposit: 10500.0

Balance after withdrawal: 8500.0

5. Create a Tea class in Java that includes the following methods:

a)prepareTea() - a method that prepares a basic tea with hot water and tea leaves.

b)addMilk() - a method that adds milk to the basic tea.

c)addSugar() - a method that adds sugar to the basic tea.

Program:

public class Tea {

protected String tea;

public Tea() {

this.tea="";

}

public void prepareTea() {

this.tea += "Hot water and tea leaves added.\n"; // \n added to print next output in next line

}

public void addMilk() {

this.tea += "Milk added.\n";

}

public void addSugar() {

this.tea += "Sugar added.\n";

}

public String getTea() {

return this.tea;

}

public static void main(String[] args) {

Tea t1= new Tea();

t1.prepareTea();

t1.addMilk();

t1.addSugar();

System.***out***.println(t1.getTea());

}

}

Output:

Hot water and tea leaves added.

Milk added.

Sugar added.

﻿

6. Create three subclasses of the Tea class: Black Tea, GreenTea, and HerbalTea. Each subclass should override the prepareTea() method to prepare the specific type of tea (black tea, green tea, or herbal tea) with appropriate ingredients and brewing times.

Program:

public interface Main {

public class BlackTea extends Tea {

public void prepareTea() {

this.tea += "Hot water and black tea leaves added.\n" +"Brew for 5 mins.\n";

}

}

public class GreenTea extends Tea {

public void prepareTea() {

this.tea += "Hot water and green tea leaves added.\n"+"Brew for 3 mins.\n";

}

}

public class HerbalTea extends Tea{

public void prepareTea() {

this.tea+= "Hot water and herbal tea leaves added.\n"+"Brew for 10 mins.\n";

}

}

public static void main(String[] args) {

BlackTea bt= new BlackTea();

GreenTea gt= new GreenTea();

HerbalTea ht= new HerbalTea();

bt.prepareTea();

bt.addMilk();

bt.addSugar();

System.***out***.println("Black Tea preparation: \n"+bt.getTea());

gt.prepareTea();

System.***out***.println("Green Tea preparation: \n"+gt.getTea());

ht.prepareTea();

ht.addMilk();

ht.addSugar();

System.***out***.println("Herbal Tea preparation: \n"+ht.getTea());

}

}

Output:

Black Tea preparation:

Hot water and black tea leaves added.

Brew for 5 mins.

Milk added.

Sugar added.

Green Tea preparation:

Hot water and green tea leaves added.

Brew for 3 mins.

Herbal Tea preparation:

Hot water and herbal tea leaves added.

Brew for 10 mins.

Milk added.

Sugar added.

7. Implement polymorphism in your program by creating an array of Tea objects that includes instances of the Tea class and its subclasses.

Program:

public interface Polymorphism {

public static void main(String[] args) {

Tea[] teas = new Tea[4];

teas[0] = new Tea();

teas[1] = new BlackTea();

teas[2] = new GreenTea();

teas[3] = new HerbalTea();

for (Tea tea : teas) {

tea.prepareTea();

System.***out***.println(tea.getTea());

}

}

}

Output:

Hot water and tea leaves added.

Hot water and black tea leaves added.

Brew for 5 mins.

Hot water and green tea leaves added.

Brew for 3 mins.

Hot water and herbal tea leaves added.

Brew for 10 mins.