Task No 10

https://medium.com/@vghadigaokar/task-10-guvi-54ccfe304d5b

Question No 1:

// Person class created with two attributes String name and int age  
 public class Person {  
 private String name;  
 private int age;  
  
//Constructer is created  
 public Person(String name, int age){  
 this.name = name;  
 this.age = age;  
 }  
  
 //created getter methods for the created attributes  
 public String getName(){  
 return name;  
 }  
 public int getAge(){  
 return age;  
 }  
  
  
 public static void main(String[] args) {  
 //we have created person1 object and passed arguments  
 Person person1 = new Person("Vinayak Ghadigaokar",27);  
 //printing the attributes using getter methods  
 System.*out*.println(person1.getName());  
 System.*out*.println(person1.getAge());  
 }  
}

Question No 2:

// Employee class created with four attributes int id, String firstname, String lastname, int salary;  
public class Employee {  
 private int id;  
 private String firstName;  
 private String lastName;  
 private int salary;  
// Constructor is created with arguments of firstname lastname and salary id  
 public Employee(int id,String firstName,String lastName,int salary){  
 this.firstName = firstName;  
 this.id= id;  
 this.lastName = lastName;  
 this.salary=salary;  
 }

// getting the id  
 public int getId() {  
 return id;  
 }  
 // getting the firstname  
 public String getFirstName() {  
 return firstName;  
 }  
 // getting the firstname  
 public String getLastName() {  
 return lastName;  
 }  
 //getname = firstname + lastname  
 public String getName(){  
 return firstName+ " "+lastName;  
 }  
 //getting the salary  
 public int getSalary() {  
 return salary;  
 }  
 //method for setting salary  
 public void setSalary(int salary) {  
 this.salary = salary;  
 }  
 //salary\*12  
 public int getAnnualSalary(){  
 return salary \* 12;  
 }  
  
 //increasing the salary by percent parameter and return the new salary  
 public int raiseSalary(int percent){  
 int raisedSalary = (salary \* percent )/100;  
 return salary+raisedSalary;  
 }  
  
 //toString id firstname lastname salary  
 public String toString(){  
  
 return id+" "+firstName+" "+lastName+" "+salary;  
 }  
  
 public static void main(String[] args) {  
 //we have created employee1 and employee2 object and passed arguments  
 Employee employee1 = new Employee(1,"Vinayak","Ghadigaokar",150000);  
 Employee employee2 = new Employee(2,"Tejas","Ghadigaokar",500000);  
 //printing test  
 System.*out*.println(employee1.getName());  
 System.*out*.println(employee1.getAnnualSalary());  
 System.*out*.println(employee1.raiseSalary(10));  
 System.*out*.println(employee1.toString());  
 System.*out*.println(employee2.toString());  
 //employee2 to set salary 5000000  
 employee2.setSalary(5000000);  
 System.*out*.println(employee2.toString());  
 }  
}

Question No 3:

public class circle {  
 private double radius;  
 private double pi;  
 // Constructor with no arguments  
 public circle() {  
 this.radius = 0.0;  
 }  
 // Constructor with two arguments  
 public circle(double radius, double pi) {  
 this.pi = pi;  
 this.radius = radius;  
 }  
  
 // Method to calculate the circumference of the circle  
 public double calculateCircumference() {  
 return 2 \* pi \* radius;  
 }  
  
 // Getter and Setter methods for the radius  
 public double getRadius() {  
 return radius;  
  
 }  
  
 public void setRadius(double radius) {  
 this.radius = radius;  
 }  
  
 public double getPi(){  
 return pi;  
 }  
 public void setPi(double pi) {  
 this.pi = pi;  
 }  
  
 // Example usage of the Circle class  
 public static void main(String[] args) {  
 // Create a Circle object using the no-argument constructor  
 circle circle1 = new circle();  
 circle1.setRadius(5.0);  
 circle1.setPi(3.14);  
 System.*out*.println("Circle 1 Circumference: " + circle1.calculateCircumference());  
  
 // Create a Circle object using the constructor with two arguments  
 circle circle2 = new circle(3.0,3.14);  
 System.*out*.println("Circle 2 Circumference: " + circle2.calculateCircumference());  
 }  
}

Question No 4:

import java.util.Scanner;  
//Account class created  
public class Account {  
 private long actno;  
 private float balance;  
  
 Scanner sc = new Scanner(System.*in*);  
//getter and setter method for balance  
 public void setBalance(float balance) {  
 this.balance = balance;  
 }  
  
 public float getBalance() {  
 return balance;  
 }  
// Getter and setter method of act no  
 public void setActno(long actno) {  
 this.actno = actno;  
 }  
  
 public long getActno() {  
 return actno;  
 }  
  
 //constructor created with no args  
 public Account() {  
 this.actno = 0L;  
 }  
 // Constructor created with two args actno and balance  
 public Account( long actno, float balance){

this.actno = actno;

this.balance = balance;  
 }  
  
 //method to deposit money  
 public void deposit() {  
 float amt;  
 System.*out*.println("Enter the amount you want to deposit: ");  
 amt = sc.nextFloat();  
 balance = balance + amt;  
 }  
 //method to withdraw money  
 public void withdrawal() {  
 float amt;  
 System.*out*.println("Enter the amount you want to withdraw: ");  
 amt = sc.nextFloat();  
 if (balance >= amt) {  
 balance = balance - amt;  
 System.*out*.println("Balance after withdrawal: " + balance);  
 } else {  
 System.*out*.println("Your balance is less than " + amt + "\tTransaction failed...!!" );  
 }  
 }  
  
 public static void main(String[] args) {  
  
 Account account1 = new Account();  
 account1.setActno(101);  
 account1.setBalance(100000f);  
 account1.deposit();  
 System.*out*.println(account1.getBalance());  
 account1.withdrawal();  
 System.*out*.println("Ac No/"+account1.getActno()+" Balance: "+ account1.getBalance());  
  
 Account account2 = new Account(102,100000);  
 System.*out*.println("Ac No/"+account2.getActno()+" Balance: "+ account2.getBalance());  
  
  
 }  
}

Question No 5

//question no 5  
// Tea class  
public class Tea {  
 private boolean isPrepared;  
 private boolean hasMilk;  
 private boolean hasSugar;  
  
 public Tea() {  
 this.isPrepared = false;  
 this.hasMilk = false;  
 this.hasSugar = false;  
 }  
// creating a method to prepare tea.  
 public void prepareTea() {  
 if (!isPrepared) {  
 System.out.println("Preparing basic tea with hot water and tea leaves.");  
 isPrepared = true;  
 } else {  
 System.out.println("Tea has already been prepared.");  
 }  
 }  
// creating a method to add milk  
 public void addMilk() {  
 if (isPrepared) {  
 if (!hasMilk) {  
 System.out.println("Adding milk to the tea.");  
 hasMilk = true;  
 } else {  
 System.out.println("Milk has already been added to the tea.");  
 }  
 } else {  
 System.out.println("Please prepare the tea first.");  
 }  
 }  
//creating a method to add sugar  
 public void addSugar() {  
 if (isPrepared) {  
 if (!hasSugar) {  
 System.out.println("Adding sugar to the tea.");  
 hasSugar = true;  
 } else {  
 System.out.println("Sugar has already been added to the tea.");  
 }  
 } else {  
 System.out.println("Please prepare the tea first.");  
 }  
 }  
}  
//main  
public class Main {  
 public static void main(String[] args) {  
 Tea tea = new Tea();  
 tea.prepareTea(); // Prepare basic tea  
 tea.addMilk(); // Add milk to the tea  
 tea.addSugar(); // Add sugar to the tea  
 tea.addMilk(); // Try adding milk again (already added)  
 tea.prepareTea(); // Try preparing tea again (already prepared)  
 }  
}  
//output  
Preparing basic tea with hot water and tea leaves.  
Adding milk to the tea.  
Adding sugar to the tea.  
Milk has already been added to the tea.  
Tea has already been prepared.

Question No 6

//question no 6  
public class Tea {  
 protected boolean isPrepared;  
 protected boolean hasMilk;  
 protected boolean hasSugar;  
  
 public Tea() {  
 this.isPrepared = false;  
 this.hasMilk = false;  
 this.hasSugar = false;  
 }  
  
 public void prepareTea() {  
 if (!isPrepared) {  
 System.out.println("Preparing basic tea with hot water and tea leaves.");  
 isPrepared = true;  
 } else {  
 System.out.println("Tea has already been prepared.");  
 }  
 }  
  
 public void addMilk() {  
 if (isPrepared) {  
 if (!hasMilk) {  
 System.out.println("Adding milk to the tea.");  
 hasMilk = true;  
 } else {  
 System.out.println("Milk has already been added to the tea.");  
 }  
 } else {  
 System.out.println("Please prepare the tea first.");  
 }  
 }  
  
 public void addSugar() {  
 if (isPrepared) {  
 if (!hasSugar) {  
 System.out.println("Adding sugar to the tea.");  
 hasSugar = true;  
 } else {  
 System.out.println("Sugar has already been added to the tea.");  
 }  
 } else {  
 System.out.println("Please prepare the tea first.");  
 }  
 }  
}  
  
//black tea subclass  
public class BlackTea extends Tea {  
 @Override  
 public void prepareTea() {  
 if (!isPrepared) {  
 System.out.println("Preparing black tea with hot water and black tea leaves.");  
 System.out.println("Brewing time: 5 minutes");  
 isPrepared = true;  
 } else {  
 System.out.println("Black tea has already been prepared.");  
 }  
 }  
}  
//green tea subclass  
public class GreenTea extends Tea {  
 @Override  
 public void prepareTea() {  
 if (!isPrepared) {  
 System.out.println("Preparing green tea with hot water and green tea leaves.");  
 System.out.println("Brewing time: 3 minutes");  
 isPrepared = true;  
 } else {  
 System.out.println("Green tea has already been prepared.");  
 }  
 }  
}  
  
//herbal tea subclass  
public class HerbalTea extends Tea {  
 @Override  
 public void prepareTea() {  
 if (!isPrepared) {  
 System.out.println("Preparing herbal tea with hot water and herbal tea ingredients.");  
 System.out.println("Brewing time: 7 minutes");  
 isPrepared = true;  
 } else {  
 System.out.println("Herbal tea has already been prepared.");  
 }  
 }  
}  
  
//main  
  
public class Main {  
 public static void main(String[] args) {  
 BlackTea blackTea = new BlackTea();  
 blackTea.prepareTea(); // Prepare black tea  
 blackTea.addMilk(); // Add milk to black tea  
 blackTea.addSugar(); // Add sugar to black tea  
  
 GreenTea greenTea = new GreenTea();  
 greenTea.prepareTea(); // Prepare green tea  
 greenTea.addSugar(); // Add sugar to green tea  
  
 HerbalTea herbalTea = new HerbalTea();  
 herbalTea.prepareTea(); // Prepare herbal tea  
 herbalTea.addMilk(); // Try adding milk to herbal tea (not applicable)  
 }  
}  
  
//output  
  
Preparing black tea with hot water and black tea leaves.  
Brewing time: 5 minutes  
Adding milk to the tea.  
Adding sugar to the tea.  
Preparing green tea with hot water and green tea leaves.  
Brewing time: 3 minutes  
Adding sugar to the tea.  
Preparing herbal tea with hot water and herbal tea ingredients.  
Brewing time: 7 minutes  
Herbal tea has already been prepared.

Question No 7:

//question no 7  
//main  
public class Main {  
 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();  
 tea.addMilk();  
 tea.addSugar();  
 System.out.println("--------------------");  
 }  
 }  
}  
  
//output  
  
Preparing basic tea with hot water and tea leaves.  
Adding milk to the tea.  
Adding sugar to the tea.  
--------------------  
Preparing black tea with hot water and black tea leaves.  
Brewing time: 5 minutes  
Adding milk to the tea.  
Adding sugar to the tea.  
--------------------  
Preparing green tea with hot water and green tea leaves.  
Brewing time: 3 minutes  
Adding sugar to the tea.  
--------------------  
Preparing herbal tea with hot water and herbal tea ingredients.  
Brewing time: 7 minutes  
Herbal tea has already been prepared.  
--------------------