CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

CHANDUBHAI S PATEL INSTITUTE OF TECHNOLOGY

Name: Abhi

Bhimani

ID:- 21CE013

CSPIT – CE

GitHub Link:- https://github.com/abhii14758/-JAVA-PRACTICAL-FILE-2

	Practical-2
Practical 2.1	Design a class named Circle containing following attributes and behavior. • One double data field named radius. The default value is 1. • A no-argument constructor that creates a default circle. • A Single argument constructor that creates a Circle with the specified radius. • A method named getArea() that returns area of the Circle. • A method named getPerimeter() that returns perimeter of it.
CODE	

```java
import java.util.*;

class circle{
    double r;

    circle()
    {
        r = 1;
    }
    circle(double a)
    {
        r = a;
    }
    public void getarea()
    {
        System.out.println("Area Of Circle = "+2*3.14*r*r);
    }
    public void getperimiter()
    {
        System.out.println("Perimiter Of Circle = "+2*3.14*r);
    }
}
```

output

```
PROBLEMS 8   OUTPUT   TERMINAL   JUPYTER   DEBUG CO
in'
Enter Value of Radius Here : 10
628.0
62.800000000000004
6.28
Enter Value of Radius Here : 10
Area Of Circle = 628.0
Perimiter Of Circle = 62.800000000000004
Area Of Circle = 6.28
Perimiter Of Circle = 6.28
PS C:\Users\HP\Desktop\Java Assignment>
```

Practical 2.2

Design a class named Account that contains:
• A private int data field named id for the account (default 0).
• A private double data field named balance for the account (default 500₹).
• A private double data field named annualInterestRate that stores the current interest rate (default 7%). Assume all accounts have the same interest rate.
• A private Date data field named dateCreated that stores the date when the account was created.
• A no-arg constructor that creates a default account.
• A constructor that creates an account with the specified id and initial balance.
• The accessor and mutator methods for id, balance, and annualInterestRate.
• The accessor method for dateCreated.
• A method named getMonthlyInterestRate() that returns the monthly interest rate.
• A method named getMonthlyInterest() that returns the monthly interest.
• A method named withdraw that withdraws a specified amount from the account.
• A method named deposit that deposits a specified amount to the account.

CODE

```java
import java.util.*;
import java.sql.Date;
class account{

    int id;
    double balance;
    double annual_interest=7;
    String date = "01-08-2022";
    double monthly_interest;
    double monthly_interest_rate;

    account()
    {
        id = 0;
        balance = 500;
    }
    account(int a,double b)
    {
```

```
        id = a;
        balance = b;
    }
    public double getAnnual_interest() {
        return annual_interest;
    }
    public double getBalance() {
        return balance;
    }
    public Date getDate() {
        return date;
    }
    public int getId() {
        return id;
    }
    public void setAnnual_interest(double annual_interest) {
        this.annual_interest = annual_interest;
    }
    public void setBalance(double balance) {
        this.balance = balance;
    }
    public void setId(int id) {
        this.id = id;
    }

    public void interest_rate()
    {
        monthly_interest_rate=annual_interest/12;
        System.out.println("Monthly Interest Rate =
"+monthly_interest_rate);
    }
    public void interest()
    {
        monthly_interest = (monthly_interest_rate*balance)/100;
        System.out.println("Monthly interest =
"+monthly_interest);
    }
    public void widraw(double c)
    {
        balance = balance - c;
        System.out.println("New Blanace = "+balance);
    }
    public void deposite(double d)
    {
        balance = balance + d;
        System.out.println("New Balance = "+balance);
    }
    public void details()
    {
        System.out.println("Account id = "+id);
        System.out.println("Account Balance = "+balance);
        interest_rate();
        interest();
```

```
        System.out.println(date);
    }

}
```

Main program

```
//this program is prepared by 21ce013 Abhi Bhimani
//Design a class named Account that contains:
//• A private int data field named id for the account (default
0).
//• A private double data field named balance for the account
(default 500₹).
//• A private double data field named annualInterestRate that
stores the current interest rate (default 7%). Assume all
accounts have the same interest rate.
//• A private Date data field named dateCreated that stores
the date when the account was created.
//• A no-arg constructor that creates a default account.
//• A constructor that creates an account with the specified
id and initial balance.
//• The accessor and mutator methods for id, balance, and
annualInterestRate.
//• The accessor method for dateCreated.
//• A method named getMonthlyInterestRate() that returns the
monthly interest rate.
//• A method named getMonthlyInterest() that returns the
monthly interest.
//• A method named withdraw that withdraws a specified amount
from the account.
//• A method named deposit that deposits a specified amount to
the account.
// // GITHUB LINK : https://github.com/abhii14758/-JAVA-
PRACTICAL-FILE-2
import java.util.*;
public class PR_2_Main{
    public static void main(String[] args) {

        int c;
        double d,w;
        Scanner s = new Scanner(System.in);
        account a = new account(1,10000);
```

```
        System.out.println("Enter 1 For Deposite ");
        System.out.println("Enter 2 For Widraw ");
        System.out.println("Enter 3 For Account Details ");
        c = s.nextInt();

        if(c ==1)
        {
            System.out.print("Enter Ammount To Deposite: ");
            d = s.nextDouble();
            a.deposite(d);
            a.details();
        }
        else if(c==2)
        {
            System.out.print("Enter Ammount To Widraw: ");
            w = s.nextDouble();
            a.widraw(w);
            a.details();
        }
        else if(c==3)
        {
            a.details();
        }
    }
}
```

Output

```
\Code\User\workspaceStorage\fa7a27b04fd424668998
in'
Enter 1 For Deposite
Enter 2 For Widraw
Enter 3 For Account Details
1
Enter Ammount To Deposite: 200
New Balance = 10200.0
Account id = 1
Account Balance = 10200.0
Monthly Interest Rate = 0.5833333333333334
Monthly interest = 59.5
01-08-2022
PS C:\Users\HP\Desktop\Java Assignment>
```

Practical 2.3

Use the Account class created as above to simulate an ATM machine. Create 10 accounts with id AC001…..AC010 with initial balance 300₹. The system prompts the users to enter an id. If the id is entered incorrectly, ask the user to enter a correct id. Once an id is accepted, display menu with multiple choices. 1. Balance inquiry 2. Withdraw money [Maintain minimum balance 300₹] 3. Deposit money 4. Money Transfer 5. Create Account 6. Deactivate Account 7. Exit Hint: Use ArrayList, which is can shrink and expand with compared to Array

CODE

```java
class ATM {
    private static int count;
    private final String id;
    private double balance;

    //method which returns ID
    public String getId() {
        return id;
    }

    //method which returns balance
    public double getBalance() {
        return balance;
    }

    //default constructor
    public ATM() {
        count++;
        if (count < 10) {
            id = "AC00" + (count);
        } else {
            id = "AC0" + (count);
        }
        balance = 300;
    }

    //withsraw method
    public void withdraw(double money) {
        if (balance - money >= 300) {
            balance -= money;
```

```
            System.out.println(money + " Rs successfully
withdrawn.");
            System.out.println("Remaining Balance is : " +
balance);
        } else {
            System.out.println("Insufficient balance to
withdraw the amount.");
        }
    }

    //deposite method
    public void deposit(double amount) {
        balance += amount;
        System.out.println(amount + "Rs deposited to your
account.");
        System.out.println("Current Balance is : " + balance);
    }

    //method for transfering money
    public void MoneyTransfer(ATM obj, double amount) {
        if (balance - amount >= 300) {
            balance -= amount;
            obj.balance += amount;
            System.out.println(amount + " Rs successfully
Transferred.");
            System.out.println("Remaining Balance is : " +
balance);
        }
        else {
            System.out.println("Insufficient balance to
transfer the amount.");
        }
    }

}
```

MAIN PROGRAM

```
// this program is prepared by 21ce013 Abhi Bhimani
// Use the Account class created as above to simulate an ATM
machine.
// Create 10 accounts with id AC001…..AC010 with initial
balance 300₹.
// The system prompts the users to enter an id.
// If the id is entered incorrectly, ask the user to enter a
correct id.
// Once an id is accepted, display menu with multiple choices.
// 1. Balance inquiry
// 2. Withdraw money [Maintain minimum balance 300₹]
// 3. Deposit money
// 4. Money Transfer
// 5. Create Account
// 6. Deactivate Account
// 7. Exit Hint:
```

```
// UseArrayList, which is can shrink and expand with compared
to Array
// GITHUB LINK : https://github.com/abhii14758/-JAVA-
PRACTICAL-FILE-2

import java.util.*;

public class PR_3_Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        //declare variable as given
        String id = "";
        String id2 = "";
        boolean flag = true;
        int choice;
        double amt;
        //sreate arraylist for 10 ID
        ArrayList<ATM> people = new ArrayList<ATM>();
        for (int i = 1; i <= 10; i++) {
            people.add(new ATM());
        }
        System.out.print("Enter Your Account Number : ");
        id = sc.next();
        int userNumber = userID(id, people);

        //choice for switch case
        while (flag) {
            System.out.println();
            System.out.println("Make a choice.....");
            System.out.println("1.Balance inquiry ");
            System.out.println("2.Withdraw money ");
            System.out.println("3.Deposit money");
            System.out.println("4.Money Transfer ");
            System.out.println("5.Create Account ");
            System.out.println("6.Deactivate Account");
            System.out.println("7.Exit ");
            choice = sc.nextInt();

            //switch case for above condition
            switch (choice) {
                case 1 -> {
                    System.out.println("Account Number : " +
id);
                    System.out.println("Current Balance : " +
people.get(userNumber).getBalance());
                }
                case 2 -> {
                    System.out.print("Enter Amount To Withdraw
: ");
                    amt = sc.nextDouble();
                    people.get(userNumber).withdraw(amt);
                }
                case 3 -> {
```

```
                    System.out.print("Enter Amount To Deposit
: ");
                    amt = sc.nextInt();
                    people.get(userNumber).deposit(amt);
                }
                case 4 -> {
                    System.out.print("Enter Account Number To
Transfer Money :");
                    id2 = sc.next();
                    int u2 = userID(id2, people);
                    System.out.print("Enter Amount To Transfer
: ");
                    amt = sc.nextInt();
                    people.get(userNumber).MoneyTransfer(peopl
e.get(u2), amt);
                }
                case 5 -> {
                    people.add(new ATM());
                    System.out.println("Account Created
Successfully.");
                    System.out.println("The New Account Number
Is :" + people.get(people.size() - 1).getId());
                }
                case 6 -> {
                    people.remove(userNumber);
                    System.out.println("Account Deleted
Successfully.");
                    flag = false;
                }
                case 7 -> flag = false;
                default -> System.out.println("Make a valid
choice..");
            }
        }

    }

    //method for show user data
    public static int userID(String id, ArrayList<ATM>people)
{
        Scanner s = new Scanner(System.in);
        int user = 10000;
        int i;
        for (i = 0; i < people.size(); i++) {
            if (id.equals(people.get(i).getId())) {
                user = i;
                break;
            }
        }
        if (i == people.size()) {
            System.out.println("No Such Account Exists.\nTry
Again..");
            System.out.print("Enter your account id :");
```

```
            id = s.next();
            return userID(id, people);
        }
        else
        return user;
    }
}
```

OUTPUT

```
Enter Your Account Number : AC001

Make a choice......
1.Balance inquiry
2.Withdraw money
3.Deposit money
4.Money Transfer
5.Create Account
6.Deactivate Account
7.Exit
2
Enter Amount To Withdraw : 100
Insufficient balance to withdraw the amount.

Make a choice......
1.Balance inquiry
2.Withdraw money
3.Deposit money
4.Money Transfer
5.Create Account
6.Deactivate Account
7.Exit
7
PS C:\Users\HP\Desktop\Java Assignment>
```

Practical 2.4

(Subclasses of Account) In Programming Exercise 2, the Account class was defined to model a bank account. An account has the properties account number, balance, annual interest rate, and date created, and methods to deposit and withdraw funds. Create two subclasses for checking and saving accounts. A checking account has an overdraft limit,but a savings account cannot be overdrawn. Draw the UML diagram for the classes and then implement them. Write a test program that creates
objects of Account, SavingsAccount, and CheckingAccount and invokestheir toString() methods

CODE

```
public class P2_4 {
    private int id=0;
    double balance=500,annualInterest=7,amount;
    String dateCreated;
    P2_4() //Here we use constructor
    {
    id=0;
    balance=50000;
    annualInterest=7;
    }
    P2_4(int i,double bal) //Here we use constructor
    {
    id=i;
    balance=bal;

    }
```

```
    void setdata(int i,double bal,double aInt,String dt)
    {
    id=i;
    balance=bal;
    annualInterest=aInt;
    dateCreated=dt;
    }
    int getId() //Here we use getter
    {
    return id;
    }
    double getBal() //Here we use getter
    {
    return balance;
    }
    double getAnn() //Here we use getter
    {
    return annualInterest;
    }
    double getMonthlyInterestRate() //Here we use getter
    {
    return (annualInterest*100)/12;
    }
    double getMonthlyInterest() //Here we use getter
    {
    return balance*(annualInterest*100)/12;
    }
    String getDt() //Here we use getter
    {
    return dateCreated;
    }
    void withdraw(double amount)
    {
    balance-=amount;
    if(balance>0)
    System.out.println("The balance left after withdrawal of
Rs."+amount+" is Rs."+balance);
    else
    System.out.println("Withdrawal of Rs."+amount+" is not
possible!!");
    }
    void deposit(double amount)
    {
    balance+=amount;
    System.out.println("The balance left after deposit of
Rs."+amount+" is Rs."+balance);
    }
   }
   class SavingAccount extends P2_4 //Here we make a new class
for more bank details.
   {
    SavingAccount(double a)
    {
```

```
    amount=a;
    balance-=amount;
    }
    public String toString()
    {
    if(balance>=3000) //Here we use if else to check balance
left after widrawal and for minimum balance required
    {
    return "The balance left after withdrawal of Rs."+amount+"
is Rs. "+balance;
    }
    else
    {
    return "Beyond1 Over Draft Limit Not CE251-Java
Programming 21CE013 Abhi Bhimani Possible!!\nMinimum balance
of Rs. 3000 is required.";
    }
    }
   }
   class ChkAccount extends P2_4 //Here we make a class for
check account details
   {
    ChkAccount(double am)
    {
    amount = am;
    balance-=amount;
    }
    public String toString()
    {
    System.out.println("Withdrawal Successful!!");
    return "Now the balance left is Rs."+balance+" after the
withdrawal of Rs."+amount;
    }
   }
```

MAIN PROGRAM

```
// this program is prepared by 21ce013 Abhi Bhimani
//(Subclasses of Account) In Programming Exercise 2, the
Account class was defined to model a bank account.
//An account has the properties account number, balance,
annual interest rate, and date created, and methods to
//deposit and withdraw funds. Create two subclasses for
checking and saving accounts. A checking account has
//an overdraft limit, but a savings account cannot be
overdrawn. Draw the UML diagram for the classes and then
//implement them. Write a test program that creates objects of
Account, SavingsAccount, and CheckingAccount
//and invokes their toString() methods.
// GITHUB LINK : https://github.com/abhii14758/-JAVA-
PRACTICAL-FILE-2

public class P2_4Main {
    public static void main(String[] args) {
        P2_4 a1 = new P2_4();
```

```
        P2_4 a2 = new P2_4(123456, 100000);
        a2.setdata(1289031, 100000, 5.6, "12-5-2020");
        System.out.println("Account Details:\n");
        System.out.println("Balance :" + a2.getBal());
        System.out.println("Annual Interest :" + a2.getAnn());
        System.out.println("Monthly Interest Rate:" +
a2.getMonthlyInterestRate());
        System.out.println("Monthly Interest :" +
a2.getMonthlyInterest());
        System.out.println("Account was created on " +
a2.getDt());
        a2.withdraw(12000);
        a2.deposit(15000);
        System.out.print("-----------------------\n");
        SavingAccount a = new SavingAccount(900); // Make the
object to pass the argument
        ChkAccount b = new ChkAccount(1000); // Make the
object to pass the argument
        System.out.println("For Saving Account:\n");
        System.out.println(a);
        System.out.print("-----------------------\n");
        System.out.println("For Checking Account:\n");
        System.out.println(b);
        System.out.println("Created By 21ce013 Abhi Bhimani");
    }
}
```

OUTPUT

```
Account Details:

Balance :100000.0
Annual Interest :5.6
Monthly Interest Rate:46.666666666666664
Monthly Interest :4666666.666666667
Account was created on 12-5-2020
The balance left after withdrawal of Rs.12000.0 is Rs.88000.0
The balance left after deposit of Rs.15000.0 is Rs.103000.0
-----------------------
For Saving Account:

The balance left after withdrawal of Rs.900.0 is Rs. 49100.0
-----------------------
For Checking Account:

Withdrawal Successful!!
Now the balance left is Rs.49000.0 after the withdrawal of Rs.1000.0
Created By 21ce013 Abhi Bhimani
```

Practical 2.5

Develop a Program that illustrate method overloading concept.

CODE

```
public class PR_5_Sub
{
    //mrthod for show student information
```

```
    public static void information(String roll_no,String
name,String branch)
    {
        System.out.println("The name of the Student
is  :"+name);
        System.out.println("The roll no of the student
is  "+roll_no);
        System.out.println(name+ " studies in "+branch);
    }

    //override information method for faculty
    public static void information(String name ,String degree,
String faculty_of,int experience,long salary)
    {
        System.out.println("Name of the professor
is  :"+name);
        System.out.println(name+" has compleated "+degree);
        System.out.println(name+ "have "+experience+" years of
teaching");
        System.out.println(name+ " teaches in "+faculty_of);
        System.out.println(name+ " has been given "+salary+"
rs of salary per year");
    }

}
```

MAIN PROGRAM

```
// this program is prepared by 21ce013 Abhi Bhimani
//Develop a Program that illustrate method overloading
concept.
// GITHUB LINK : https://github.com/abhii14758/-JAVA-
PRACTICAL-FILE-2

import java.util.Scanner;

public class PR_5_Main{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        //declare variable as required
        String
name_std,roll_no_std,branch_std,faculty_of_pro,name_pro,degree
_pro;
        int experience;
        long salary;

        //input fr student
        System.out.println("Taking information of
student  :");
        System.out.print("Name of student  :");
        name_std = s.next();
        System.out.print("Roll no of student  :");
        roll_no_std = s.next();
        System.out.print("Branch student studies in  :");
        branch_std = s.next();
```

```
        System.out.println("--------------------------------
-------------------------------");

        //input for faculty
        System.out.println("Taking information for
faculty  :");
        System.out.print("Name of professor  :");
        name_pro = s.next();
        System.out.print("Degree of professor  :");
        degree_pro = s.next();
        System.out.print("Experience of teaching  :");
        experience = s.nextInt();
        System.out.print("Teaches in  :");
        faculty_of_pro = s.next();
        System.out.print("His/her salary  :");
        salary = s.nextLong();
        System.out.println("--------------------------------
-------------------------------");

        //method information call for show data of student
        System.out.println("The given information of student
is  :");
        PR_5_Sub.information(roll_no_std, name_std,
branch_std);
        System.out.println("--------------------------------
-------------------------------");

        //method information call for show data of faculty
member
        System.out.println("The given information of professor
is  :");
        PR_5_Sub.information(name_pro, degree_pro,
faculty_of_pro, experience, salary);
    }
}
```

OUTPUT

```
Taking information of student  :
Name of student  :Abhi
Roll no of student  :21ce013
Branch student studies in  :ce
-------------------------------------------------------------
Taking information for faculty  :
Name of professor  :ronak
Degree of professor  :PHD
Experience of teaching  :15
Teaches in  :ce
His/her salary  :530000
-------------------------------------------------------------
The given information of student is  :
The name of the Student is  :Abhi
The roll no of the student is  21ce013
Abhi studies in ce
-------------------------------------------------------------
The given information of professor is  :
Name of the professor is  :ronak
ronak has compleated PHD
ronakhave 15 years of teaching
ronak teaches in ce
ronak has been given 530000 rs of salary_per year
```