Lab-5

Page No.
Date

Q Develop a java program to create a class Bank that maintains two kinds of account for its customers, one saving, other current. The saving account provides compound interest and withdrawal facilities but no cheque. Current account holders should also maintain a minimum balance, and if the balance falls below this level, a service charge is imposed

```
import java.util.Scanner;

class Account{
    String customerName;
    int accountNumber;
    String accountType;
    double balance;

    Account(String customerName, int accountNumber, String
    accountType){
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0;
    }
    void deposit(double amount)
    {
        balance += amount;
        System.out.println("Deposit succes
        Update balance:" + balance);
    }

    void displayBalance(){
        System.out.println("Account balance:" + ba
    }
```

```
    void withdraw(double amount)
    { }
}

class CurAcct extends Account{
    double minBalance, serviceCharge;

    CurAcct(String customerName, int accountNumber)
    {
        super(customerName, accountNumber, "current");
        this.minBalance = 1000;
        this.serviceCharge = 50;
    }

    @Override
    void deposit(double amount)
    {
        super.deposit(amount);
        checkMinimumBalance();
    }

    private void checkMinimumBalance()
    {
        if (balance < minBalance)
        {
            balance -= serviceCharge;
            System.out.println("Service charge
imposed. Updated balance:" + balance);
        }
    }
}

class SavAcct extends Account{
    double interestRate;
```

```
SavAct(String customerName, int accountNumber)
{
    super(customerName, accountNumber, "Saving");
    this.interestRate = 0.05;
}
@Override
void deposit(double amount)
{
    super.deposit(amount);
    computeInterest();
}
void computeInterest() {
    double interest = balance * interestRate;
    balance += interest;
    System.out.println("Interest computed
    and deposited. Updated balance:" + balance
}
void withdraw(double amount)
{
    if (amount <= balance)
    {
        balance -= amount;
        System.out.println("Withdrawal success
        .Updated balance:" + balance);
    }
    else
        System.out.println("Insufficient funds for
        withdrawal");
    }
}
public class Main{
```

```
public static void main (String[] args)
{
    Scanner sc = new Scanner(System.in);
    int ch;
    Account userAccount = null;
    System.out.println("Enter customer name:");
    String customerName = sc.next();
    System.out.println("Enter account number:");
    String accountType = sc.next();
    if (accountType.equals("savings")){
        userAccount = new SavAct(customerName,
        accountNumber);
    }
    else if (accountType.equals("current")){
        userAccount = new CurAct(customerName,
        accountNumber);
    }
    else{
        System.out.println("Invalid account type.
        Existing program");
        sc.close();
        System.exit(0);
    }
    while(true){
        System.out.println("---MENU---\n1.
Deposit\n2.Withdraw\n3. Display Account
Details\n4. Exit");
        System.out.println("Enter your choice:");
        ch = sc.nextInt();

    switch(ch){
```

```
case 1: System.out.println("Enter dep. amt");
        double depositAmount = sc.nextDouble();
        userAccount.deposit(depositAmount);
        break;

case 2: System.out.println("Enter withdrawal amount");
        double withdrawalAmount = sc.nextDouble();
        userAccount.withdraw(withdrawalAmount);
        break;

case 3: System.out.println("Customer name:" + userAccount.
        customerName);
        System.out.println("Account number:" +
        userAccount.accountNumber);
        System.out.println("Type of Account:"
        + userAccount.accountType);
        System.out.println("Balance:" + userAccount
        .balance);
        break;

case 4: System.out.println("Exiting program.");
        sc.close();
        System.exit(0);

default: System.out.println("Invalid choice.
         Please enter a valid option");
}
}
}
```

Output

Enter customer name: Shruti
Enter account number: 1
Enter the type of account: saving
---MENU---
1. Deposit
2. Withdraw
3. Display Account Details
4. Exit
Enter your choice: 1
Enter deposit amount: 1000
Deposit successful
Interest computed and deposited: 1050
Enter your choice: 2
Enter withdrawal amount: 500
Withdrawal successful Balance: 550
Enter your choice: 3
Customer name: Shruti
Account number: 2
Type of account: Savings
Balance: 550

Page No.
Date

Q. Write a Java program to create a generic class stack which hold 5 integers and 5 double values

```
class GenericStack<T>{
    private Object[] stackArray;
    private int top=-1;
    private static final int MAX_SIZE=5;

    public GenericStack()
    {
        stackArray=new Object[MAX_SIZE];
    }

    public void push(T value){
        if(top<MAX_SIZE-1)
            stackArray[++top]=value;
        else
            System.out.println("Stackfull");
    }
    @SuppressWarnings("unchecked")
    public T pop(){
        if(top>=0)
            return(T)stackArray[top--];
        else{
            System.out.println("Empty stack");
            return null;
        }
    }
    public boolean isEmpty(){
        return top==-1;
    }
```

```
public boolean isFull()
{
    return top == MAX_SIZE - 1;
}

class Main {
    public static void main(String[] args)
    {
        GenericStack<Integer> integerStack = new GenericStack();
        GenericStack<Double> doubleStack = new GenericStack();

        for (int i = 1; i <= 5; i++)
        {
            integerStack.push(i);
        }
        for (double i = 1.0; i <= 5.0; i++) {
            doubleStack.push(i);
        }
        System.out.println("Popped integers:");
        while (!integerStack.isEmpty()){
            System.out.println(integerStack.pop());
        }
        System.out.println("Popped doubles:");
        while (!doubleStack.isEmpty())
        {
            System.out.println(doubleStack.pop());
        }
    }
}
```

Output

Popped integers from the stack:

5

4

3

2

1

Popped doubles:

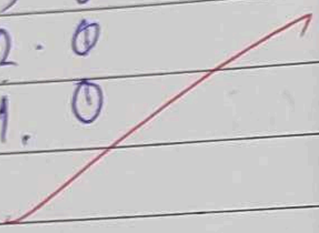
5.0

4.0

3.0

2.0

1.0



Write a Java program to create an abstract class Bird with abs. Methods fly() and makeSound(), create subclasses Eagle and Hawk that extend the Bird class and implement the respective methods to describe how each bird flies and makes a sound.

```
abstract class Bird
        public abstract void fly();
        public abstract void makeSound();
}

class Eagle extends Bird{
        @Override
        public void fly(){
                System.out.println("Eagle soars high");
        }
        @Override
        public void makeSound(){
                System.out.println("Eagle screeches loudly");
        }
}
class Hawk extends Bird{
        @Override
        public void fly(){
                System.out.println("Hawk glides gracefully");
        }
        @Override
        public void makeSound(){
                System.out.println("Hawk emits a piercing
        }
}
public class BirdDemo{
        public static void main(String[] args){
```

```
Eagle eagle = new Eagle();
Hawk hawk = new Hawk();

System.out.println("Eagle:");
eagle.fly();
Eagle.makeSound();

system.out.println("\nHawk:");
hawk.fly();
hawk.makeSound();
    }
}
```

Output

```
Eagle:
Eagle soars high.
Eagle screeches loudly

Hawk:
Hawk glides gracefully
Hawk emits a piercing cry
```

Page No.
Date

Write a Java program to demonstrate concat() for s1 = "Hello" and s2 = "world"

```
public class ConcatenationDemo{
    public static void main(String[] args)
    {
        String s1 = "Hello";
        String s2 = "world";

        String result = s1.concat(s2);
        System.out.println("Concatenated String: " + result);
    }
}
```

Output

Concatenated String: Helloworld

Write a Java program to demonstrate trim() for "Hello friends".

```
public class Trimsf
    public static void main (String [] args)
    {
        String new = "Hello friends";
        String trimmed = new trim();
        System.out.println("Old string:" + new);
        System.out.println("New string:" + trim);
    }
}
```

Output

Old string: Hello friends
New string: HelloFriends

16/01/24