Name : Rushikesh Krishna Daund

# Assignment no 1

1. Create an application for Bank to perform below operations
1. Create account
Name
Account no {Unique}
Balance
Address
2. Deposit amount (input: account no, amount)
3. Withdraw amount (input: account no, amount)
4. Display account information
5. Update account Name, Address {input account no}
Expectations:
- use OOPs concepts accordingly
- implement the validations around operations
- take inputs from user for choice then for operation
- Program should exit only when user choice matches the exit

Code :-

```
import java.util.*;

import java.text.*;

class Banking

{

	String account_no;

	String name="",address="";

	float balance;

	public void create_acc(String user_name,String ac_address)

	{

		account_no = new DecimalFormat("0000000000").format(new
Random().nextInt(999999));

		address = ac_address;

		name = user_name;

		System.out.println("account has been created ..");

	}
```

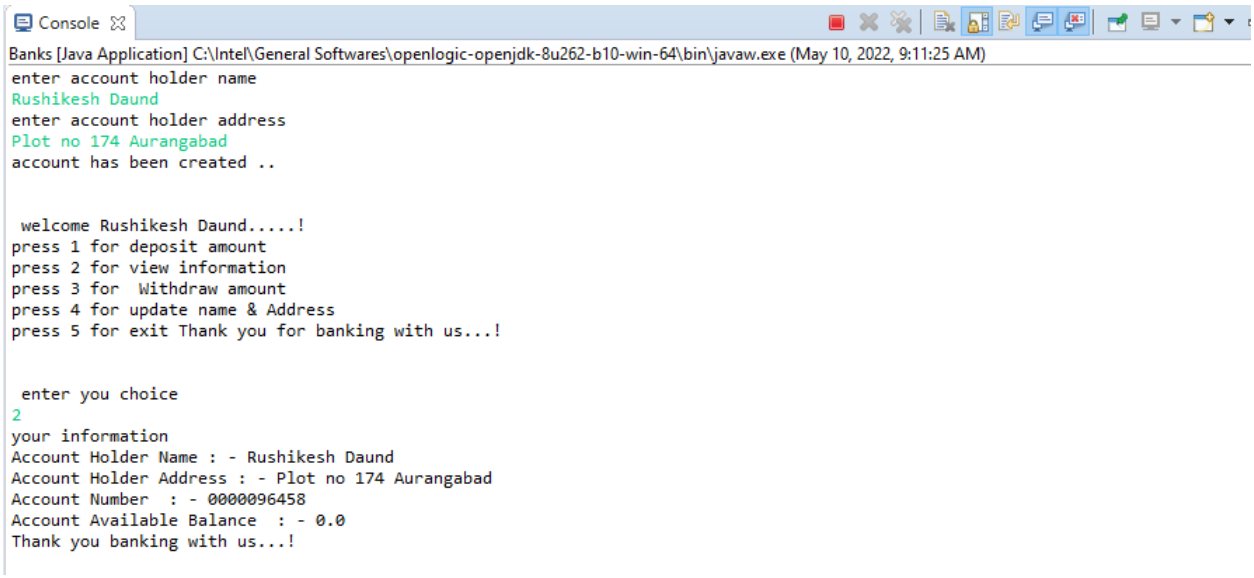
```
public void deposit(String ac_no,float credit_bal)
{
	if(account_no==ac_no)
	{
		balance = balance + credit_bal;
		System.out.println("your balance is credited Successfully ");
	}
	else
	{
		System.out.println("operation fail please try again");
	}




}
public void debit(String ac_no,float debit_bal)
{
	if(account_no==ac_no)
	{
		balance = balance - debit_bal;
		System.out.println("your balance is debited Successfully ");


	}
	else
	{
		System.out.println("operation fail please try again");
	}


}
public void display()
```

```
	{
		System.out.println("Account Holder Name : - "+name);
		System.out.println("Account Holder Address : - "+address);
		System.out.println("Account Number  : - "+account_no);
		System.out.println("Account Available Balance  : - "+balance);

	}
	public void info_update(String N1,String A1)
	{
		name = N1;
		address = A1;
	}


}
public class Banks {

	public static void main(String[] args) {
		int choice;
		float bals;
		String name,address,ac_no;
		Scanner s1 = new Scanner(System.in);
		Banking b1 = new Banking();
			System.out.println("enter account holder name ");
			name = s1.nextLine();
			System.out.println("enter account holder address");
			address = s1.nextLine();
			b1.create_acc(name, address);
```

```
do{
        System.out.println("\n \n welcome "+name+ ".....!");
        System.out.println("press 1 for deposit amount ");
        System.out.println("press 2 for view information");
        System.out.println("press 3 for  Withdraw amount ");
        System.out.println("press 4 for update name & Address");
        System.out.println("press 5 for exit Thank you for banking with us...! ");
        System.out.println("\n \n enter you choice ");
        choice = s1.nextInt();
        switch(choice)
        {
        case 1:
                System.out.println("enter account no ");
                ac_no = s1.next();
                System.out.println("enter amount ");
                bals = s1.nextFloat();
                b1.deposit(ac_no, bals);
                break;

        case 2:
                System.out.println("your information");
                b1.display();
                break;

        case 3:
                System.out.println("enter account no ");
                ac_no = s1.next();
                System.out.println("enter amount ");
                bals = s1.nextFloat();
```

```
                    b1.debit(ac_no, bals);
                    break;

                case 4:
                    System.out.println("enter name ");
                    name = s1.nextLine();
                    System.out.println("enter address ");
                    address = s1.nextLine();
                    b1.info_update(name, address);
                    break;

                case 5:
                    break;

                default:
                    System.out.println("Please enter correct choice ");
                    break;


                }

                System.out.println("Thank you banking with us...!");


        }while(true);

        }


}
```

Output:




2. Create an application to calculate difference between two input dates.
Example,
Input1: 01 March 2000 (day-Month-year)
Input2: 16 August 2021 (day-Month-year)
Output:
Difference between two input dates is 21 years 5 months 15 days.
Expectations:
- implement required validations on input

Code:-

```
import java.text.SimpleDateFormat;

import java.text.ParseException;

import java.util.Date;

import java.util.Scanner;

class DateandTime {


   static void diff(String join_date, String leave_date)

   {


      SimpleDateFormat obj = new SimpleDateFormat("dd-mm-yyyy");
```

```
    try {
        Date date1 = obj.parse(join_date);
        Date date2 = obj.parse(leave_date);
        long time_difference = date2.getTime() - date1.getTime();
        long days_difference = (time_difference / (1000*60*60*24)) % 365;
        long years_difference = (time_difference / (1000l*60*60*24*365));
        System.out.print(
            "Difference "
            + "between two dates is: ");
        System.out.println(
             ""
            + years_difference
            + " years, "
            + days_difference
            + " days"
            );
    }
    // Catch parse exception
    catch (ParseException excep) {
        excep.printStackTrace();
    }
}
// Main class
public static void main(String[] args)
{
    // Set values for both dates
         Scanner obj = new Scanner(System.in);
    String join = obj.nextLine();
    String leave  = obj.nextLine();
```

```
    diff(join, leave);
  }
}
```

3. Write a program to find length of longest consecutive sequence in array of integers?
Given an unsorted array of integers, find the length of the longest consecutive elements sequence.
Example,
Given [100, 4, 200, 1, 3, 2],
The longest consecutive elements sequence is [1, 2, 3, 4]. Return its length: 4

Code:-

```
import java.util.*;
public class SortingandSwaping {
   public static void main(String[] args) {
          int n;
          Scanner obj = new Scanner(System.in);
          System.out.println("enter size of array");
          n = obj.nextInt();
         int nums[]=new int[n];
         for(int i=0;i<n;i++)
         {
              nums[i] = obj.nextInt();
         }
              System.out.println("Original array length: "+nums.length);
              System.out.print("Array elements are: ");
        for (int i = 0; i < nums.length; i++)
         {
             System.out.print(nums[i]+" ");
         }
              System.out.println("\nThe new length of the array is:
"+Longest_sequence(nums));

    }

    public static int longest_sequence(int[] nums) {
      final HashSet<Integer> h_set = new HashSet<Integer>();
        for (int i : nums) h_set.add(i);

        int longest_sequence_len = 0;
        for (int i : nums) {
            int length = 1;
            for (int j = i - 1; h_set.contains(j); --j) {
                h_set.remove(j);
                ++length;
            }
            for (int j = i + 1; h_set.contains(j); ++j) {
                h_set.remove(j);
```

```
                ++length;
            }
            longest_sequence_len = Math.max(longest_sequence_len, length);
        }
        return longest_sequence_len;
    }
}
```

Output