OOP Assignment 2
NAME : RITABROTO GANGULY
ROLL : 001910501090

1. Design a **BankAcct** class with account number, balance and interest rate as attribute. Interest rate is same for all account. Support must be there to initialize, change and display the interest rate. Also supports are to be there to return balance and calculate interest.

```
import java.util.*;
import java.io.*;

class BankAcct{
        static int accns = 1;
        private int accn;
        private float balance;
        private static float interest = 10.0f;
        BankAcct(float balance){this.balance = balance;accn = accns++;}
        static void setInterestRate(float intrst){interest = intrst;}
        void showInterestRate(){System.out.println("Interest Rate: "+interest);}
        void showBalance(){System.out.println("Balance: "+balance);}
        float calcInterestAmount(){return balance*interest*0.01f;}
       void showAccountNumber(){System.out.println("Account Number:
"+accn);}

        public static void main(String[] args) throws Exception{
         Scanner s = new Scanner(System.in);
         System.out.print("Enter balance: ");
         BankAcct b = new BankAcct(s.nextFloat());
         b.showAccountNumber();
         b.showInterestRate();
         b.showBalance();
         System.out.println("Current interest amount is
"+b.calcInterestAmount());
         System.out.print("Set new interest rate(y/n)? ");
         if(s.next().charAt(0)=='y'){
           System.out.print("Enter new interest rate: ");
           setInterestRate(s.nextFloat());
           b.showAccountNumber();
           b.showInterestRate();
           b.showBalance();
           System.out.println("Current interest amount is
"+b.calcInterestAmount());
```

```
        }
       }
}
```

```
Enter balance: 10001.543
Account Number: 1
Interest Rate: 10.0
Balance: 10001.543
Current interest amount is 1000.1543
Set new interest rate(y/n)? y
Enter new interest rate: 12.6
Account Number: 1
Interest Rate: 12.6
Balance: 10001.543
Current interest amount is 1260.1945
```

2. Design a **Metric** class that supports Kilometre to Mile conversion with distance in Kilometre as argument and Mile to Kilometre conversion with distance in mile as argument. Assume, one Mile equals 1.5 Kilometre.

```
import java.io.*;
import java.util.Scanner;
import java.math.*;

class Metric{
        public static final BigDecimal CONV = new BigDecimal("1.5");
       static BigDecimal KmToMile(BigDecimal km,int scale){return
km.divide(CONV,scale,RoundingMode.HALF_UP);}
       static BigDecimal MileToKm(BigDecimal mile){return
mile.multiply(CONV);}
        public static void main(String[] args) throws Exception{
              System.out.print("Km to Mile(1) or Mile to Km(2)? ");
              Scanner s = new Scanner(System.in);
            int opt = s.next().charAt(0);
            System.out.print("Enter value to be converted: ");
              BigDecimal v = new BigDecimal(s.next());
            if(opt=='1'){
               int sc = v.scale();
                 System.out.println(v+" km is "+KmToMile(v,(sc+2))+" mi");
              }
            else
                 System.out.println(v+" mi is "+MileToKm(v)+" km");
       }
```

}

```
Km to Mile(1) or Mile to Km(2)? 1
Enter value to be converted: 1000
1000 km is 666.67 mi
Ritobrotos-MacBook-Air:Adv_OOP2 rgdgr8$ java Metric
Km to Mile(1) or Mile to Km(2)? 2
Enter value to be converted: 667.6667
667.6667 mi is 1001.50005 km
```

3. Take a String input that contains multiple words. Do the following: i) number of times ‘a’ appears ii) number of times “and” appears iii) whether it starts with “The” or not iv) put the String into an array of characters v) display the tokens in the String (tokens are the substrings separated by space or @ or .)

```
import java.io.*;
import java.util.*;

class P3{
	static String[] showTokens(String s){
		String[] words = s.split("\\s+|@|\\.\\s*");
		return words;
	}

	static void a(String s){
		int n = s.length();
		int ac = 0;
		for(int i = 0; i < n; i++)
		    if(s.charAt(i)=='a')ac++;
		System.out.println("Number of 'a' is "+ac+"\n");
	}

	static void and(String s){
		String[] words = showTokens(s);
	      final String and = "and";
		int andc = 0;
		for(String x : words)
		    if(x.equals(and))andc++;
		System.out.println("Number of \"and\" is "+andc+"\n");
	}

	static void The(String s){
```

```
        boolean res = false;
        String t = s.trim();
        if(t.length()>3 && t.charAt(0)=='T' && t.charAt(1)=='h' &&
t.charAt(2)=='e')
          res = true;
        System.out.println("The line starts with a \"The\": "+res+"\n");
    }

    static void showCharArray(String s){
        char[] x = s.toCharArray();
        for(char y : x)
          System.out.print(y+" ");
        System.out.println("\n");
    }

    public static void main(String[] args) throws Exception {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter line: ");
      String line  = s.nextLine();
        a(line);
        and(line);
        The(line);
        String[] words = showTokens(line);
      for(String x : words)
          System.out.println(x);
        System.out.println("");
        showCharArray(line);
    }
}
```

```
Enter line: Hi. Ritobroto@Ganguly
Number of 'a' is 1

Number of "and" is 0

The line starts with a "The": false

Hi
Ritobroto
Ganguly

H i .   R i t o b r o t o @ G a n g u l y
```

4. Consider a wrapper class for a numeric basic type. Check the support for the following: conversion from i) basic type to object ii) object to basic type iii) basic type to String iv) String (holding numeric data) to numeric object v) object to String.

```
import java.io.*;
import java.util.*;
import java.lang.reflect.*;

@SuppressWarnings("serial")
class MyException extends Exception{
	MyException(String s){super(s);}
}
@SuppressWarnings("serial")
class MyWrapper{
	private Number data;
	MyWrapper(Number data){this.data = data;}
	Number value(){Number x = data;return x;}
	void setVal(Number d){data = d;}
	static Number valueOf(String s) throws Exception {
		return (Number)Double.valueOf(s);
	}
	public String toString(){return String.valueOf(data);}

private static final Scanner s = new Scanner(System.in);
	public static void main(String[] args) throws Exception {

	Class<?> cl;//this is used during input to get any particular data type without having to know which class's valueOf(String) or parse method to call during compile time

		try{
		   System.out.print("Enter numeric class name: ");
		   String cn = s.next();
		   cn = cn.substring(0,1).toUpperCase() + cn.substring(1);
		   cl = Class.forName("java.lang."+cn);
		   System.out.print("Enter a "+cn+" value: ");
		  Method m = cl.getDeclaredMethod("valueOf",String.class);
		  MyWrapper mw = new MyWrapper((Number)m.invoke(null,s.next()));
		   System.out.println(mw.value());
		    System.out.print("Enter a new "+cn+" value to set: ");
		    mw.setVal((Number)m.invoke(null,s.next()));
		   System.out.println(mw.value());
```

```
            System.out.print("Enter a string for a "+cn+" value: ");
            Number a = MyWrapper.valueOf(s.next());
            System.out.println(a);
        }catch(InvocationTargetException e){
            throw new MyException("Incompatible type");
        }
    }
}
```

```
Enter numeric class name: integer
Enter a Integer value: 13312
13312
Enter a new Integer value to set: 344444
344444
Enter a string for a Integer value: 3333332
3333332.0
```

5. Each customer of a bank has customer id, name, and current loan amount and phone number. One can change the attributes like name, phone number. A customer may ask for loan of certain amount. It is granted provided the sum of current loan amount and asked amount does not exceed credit limit (fixed amount for all customer). A customer may be a privileged amount. For such customers credit limit is higher. Once a loan is sanctioned necessary updates should be made. Any type of customer should be able to find his credit limit, current loan amount and amount of loan he can seek. Design and implement the classes.

```
import java.io.*;
import java.util.*;

class Bank{
    private static int ids = 1;
    public static final double U_LIMIT = 20000.0;
    public static final double P_LIMIT = 100000.0;
    private HashMap<Integer,Customer> hm = new HashMap<>();
    Customer getCustomer(int id){return hm.get(id);}
    void loan(int id,double amount){
        Customer c = hm.get(id); if(c==null){System.out.println("No such
account!");return;}
        if(c.priv && (c.loan+amount)<=P_LIMIT) c.loan += amount;
```

```
            else if(!c.priv && (c.loan+amount)<=U_LIMIT) c.loan += amount;
            else System.out.println("Insufficient Credit!");
        }
        public String toString(){
            StringBuilder s = new StringBuilder("");
            for(Integer x : hm.keySet())
                s.append(hm.get(x));
            return s.toString();
        }

        private class Customer{
            private int id;
            private String name;
            private String number;
            private double loan = 0.0;
            private boolean priv;
            Customer(String name,String number,boolean priv){
                this.name=name;
                this.number=number;
                this.priv=priv;
                id=ids++;
                hm.put(id,this);
            }
            public String toString(){
                String x = ("Acc No: "+id+"  Name: "+name+"  Number:
"+number);
                double limit = priv ? P_LIMIT : U_LIMIT;
                x += ("\nCurrent Loan Amount: "+loan+"  Seekable Loan
Amount: "+(limit-loan)+"  Loan Limit: "+limit);
                x += "\n";
                return x;
            }
            void changeName(String name){this.name = name;}
            void changeNumber(String number){this.number = number;}
            int getAccNum(){return id;}
        }

        public static void main(String[] args){
            Scanner s = new Scanner(System.in);Random rand = new
Random(1);
            Bank b = new Bank();
            for(int i=0;i<5;i++) b.new
Customer(String.valueOf(i),""+i+i+i+i+i+i,rand.nextBoolean());
            System.out.println(b);
            do{
```

```
                        System.out.println("Enter account number: ");
                        Customer c = b.getCustomer(s.nextInt());
                        if(c==null){System.out.println("No such
account!");continue;}

                        System.out.println("Choose an option:");
                        System.out.println("1. Loan");
                        System.out.println("2. Show Details");
                        System.out.println("3. Change Name");
                        System.out.println("4. Change Number");
                        System.out.println("0. Exit");
                        char x = s.next().charAt(0);

                        switch(x){
                                case '1' : System.out.println("Enter Loan amount: ");
                                        b.loan(c.getAccNum(),s.nextDouble());break;

                                case '2' : System.out.println(c);break;

                                case '3' : System.out.println("Enter new Name: ");
                                        s.skip("\
\s+");c.changeName(s.nextLine());break;

                                case '4' : System.out.println("Enter new Number: ");
                                        s.skip("\
\s+");c.changeNumber(s.nextLine());break;

                                case '0' : continue;

                                default : System.out.println("Invalid option!");
                        }
                }while(true);
        }
}
```

```
Acc No: 1  Name: 0  Number: 000000
Current Loan Amount: 0.0  Seekable Loan Amount: 100000.0  Loan Limit: 100000.0
Acc No: 2  Name: 1  Number: 111111
Current Loan Amount: 0.0  Seekable Loan Amount: 20000.0  Loan Limit: 20000.0
Acc No: 3  Name: 2  Number: 222222
Current Loan Amount: 0.0  Seekable Loan Amount: 20000.0  Loan Limit: 20000.0
Acc No: 4  Name: 3  Number: 333333
Current Loan Amount: 0.0  Seekable Loan Amount: 20000.0  Loan Limit: 20000.0
Acc No: 5  Name: 4  Number: 444444
Current Loan Amount: 0.0  Seekable Loan Amount: 20000.0  Loan Limit: 20000.0

Enter account number:
1
Choose an option:
1. Loan
2. Show Details
3. Change Name
4. Change Number
0. Exit
1
Enter Loan amount:
12345666
Insufficient Credit!
Enter account number:
1
Choose an option:
1. Loan
2. Show Details
3. Change Name
4. Change Number
0. Exit
1
Enter Loan amount:
2332
Enter account number:
1
Choose an option:
1. Loan
2. Show Details
3. Change Name
4. Change Number
0. Exit
3
Enter new Name:
rg
Enter account number:
1
Choose an option:
1. Loan
2. Show Details
3. Change Name
4. Change Number
0. Exit
4
Enter new Number:
645342312
Enter account number:
1
Choose an option:
1. Loan
2. Show Details
3. Change Name
4. Change Number
0. Exit
2
Acc No: 1  Name: rg  Number: 645342312
Current Loan Amount: 2332.0  Seekable Loan Amount: 97668.0  Loan Limit: 100000.0
```

6. For every person in an institute details like name, address (consists of premises number, street, city, pin and state), phone number, e-mail id are maintained. A person is either a student or a faculty. For student roll number and course of study are also be maintained. For faculty employee id, department and specialisation are to be stored. One should be able to view the object details and set the attributes. For address, one may change it partially depending on the choice. Design and implement the classes.

```
import java.util.*;
import java.util.regex.Pattern;

class Address {
   private String premises; //premises number
   String getPremisesNumber() {return premises;}
   private String street; //street name
   String getStreetName() {return street;}
   private String city; //city name
   String getCityName() {return city;}
   private int pin; //PIN number
   int getPIN() {return pin;}
   private String state; //state name
   String getStateName() {return state;}

   Address() {
      System.out.print("Enter premises number: "); premises = P6.in.next();
      System.out.print("Enter street name: "); P6.in.skip(P6.ws); street =
P6.in.nextLine();
      System.out.print("Enter city name: "); P6.in.skip(P6.ws); city =
P6.in.nextLine();
      System.out.print("Enter PIN: "); pin = P6.in.nextInt();
      System.out.print("Enter state name: "); P6.in.skip(P6.ws); state =
P6.in.nextLine();
   }
   void update() {
      System.out.print("Premises number: "+premises+"\nDo you want to
change this value? [Y/N] ");
      if(P6.in.next().toLowerCase().charAt(0) == 'y') {System.out.print("Enter
premises number: "); premises = P6.in.next();}
      System.out.print("Street name: "+street+"\nDo you want to change this
value? [Y/N] ");
      if(P6.in.next().toLowerCase().charAt(0) == 'y') {System.out.print("Enter
street name: "); P6.in.skip(P6.ws); street = P6.in.nextLine();}
      System.out.print("City name: "+city+"\nDo you want to change this
value? [Y/N] ");
```

```
        if(P6.in.next().toLowerCase().charAt(0) == 'y') {System.out.print("Enter
city name: "); P6.in.skip(P6.ws); city = P6.in.nextLine();}
        System.out.print("PIN: "+pin+"\nDo you want to change this value? [Y/
N] ");
        if(P6.in.next().toLowerCase().charAt(0) == 'y') {System.out.print("Enter
PIN: "); pin = P6.in.nextInt();}
        System.out.print("State name: "+state+"\nDo you want to change this
value? [Y/N] ");
        if(P6.in.next().toLowerCase().charAt(0) == 'y') {System.out.print("Enter
state name: "); P6.in.skip(P6.ws); state = P6.in.nextLine();}
    }
    @Override public String toString() {return premises+" "+street+", "+city+"
"+pin+", "+state;}
}

abstract class Member {
    private String name;
    String getName() {return name;}
    private Address address;
    Address getAddress() {return address;}
    private String phone; //phone number
    String getPhoneNumber() {return phone;}
    private String email;
    String getEmail() {return email;}

    Member() {
        System.out.print("Enter name: "); P6.in.skip(P6.ws); name =
P6.in.nextLine();
        System.out.println("For address:"); address = new Address();
        System.out.print("Enter phone number: "); phone = P6.in.next();
        System.out.print("Enter e-mail address: "); email = P6.in.next();
    }
    void update() {
        System.out.print("Name: "+name+"\nDo you want to change this value?
[Y/N] ");
        if(P6.in.next().toLowerCase().charAt(0) == 'y') {System.out.print("Enter
name: "); P6.in.skip(P6.ws); name = P6.in.nextLine();}
        System.out.print("Address: "+address+"\nDo you want to change this
value? [Y/N] ");
        if(P6.in.next().toLowerCase().charAt(0) == 'y') address.update();
        System.out.print("\nPhone Number: "+phone+"\nDo you want to
change this value? [Y/N] ");
        if(P6.in.next().toLowerCase().charAt(0) == 'y') {System.out.print("Enter
phone number: "); phone = P6.in.next();}
```

```
        System.out.print("E-mail address: "+email+"\nDo you want to change
this value? [Y/N] ");
        if(P6.in.next().toLowerCase().charAt(0) == 'y') {System.out.print("Enter e-
mail address: "); email = P6.in.next();}
    }
    @Override public String toString() {
         StringBuilder s = new StringBuilder();
        s.append("Name: "+name);
        s.append("\nAddress: "+address);
        s.append("\nPhone number: "+phone);
        s.append("\nE-mail address: "+email);
         return s.toString();
    }
}

class Student extends Member {
    private final String roll; //roll number
    String getRoll() {return roll;}
    private String course; //course of study
    String getCourse() {return course;}

    Student(String roll) throws Exception {
        roll = roll.trim(); if(roll.length() == 0) throw new
IllegalArgumentException(); this.roll = roll;
        System.out.print("Enter course of study: "); P6.in.skip(P6.ws); course =
P6.in.nextLine();
    }
    @Override void update() {
        System.out.println("For student roll number "+roll);
        super.update();
        System.out.print("Course of study: "+course+"\nDo you want to change
this value? [Y/N] ");
        if(P6.in.next().toLowerCase().charAt(0) == 'y') {System.out.print("Enter
course of study: "); P6.in.skip(P6.ws); course = P6.in.nextLine();}
    }
    @Override public String toString() {
         StringBuilder s = new StringBuilder();
        s.append("Student roll number: "+roll+"\n");
        s.append(super.toString());
        s.append("\nCourse of study: "+course);
         return s.toString();
    }
}

class Faculty extends Member {
```

```
    private final String id; //employee ID
    String getEmployeeID() {return id;}
    private String dept; //department
    String getDepartment() {return dept;}
    private String special; //specialization
    String getSpecialization() {return special;}

    Faculty(String id) throws Exception {
        id = id.trim(); if(id.length() == 0) throw new IllegalArgumentException();
this.id = id;
        System.out.print("Enter department: "); P6.in.skip(P6.ws); dept =
P6.in.nextLine();
        System.out.print("Enter specialization: "); P6.in.skip(P6.ws); special =
P6.in.nextLine();
    }
    @Override void update() {
        System.out.println("For faculty employee ID: "+id);
        super.update();
        System.out.print("Department: "+dept+"\nDo you want to change this
value? [Y/N] ");
        if(P6.in.next().toLowerCase().charAt(0) == 'y') {System.out.print("Enter
department: "); P6.in.skip(P6.ws); dept = P6.in.nextLine();}
        System.out.print("Specialization: "+special+"\nDo you want to change
this value? [Y/N] ");
        if(P6.in.next().toLowerCase().charAt(0) == 'y') {System.out.print("Enter
specialization: "); P6.in.skip(P6.ws); special = P6.in.nextLine();}
    }
    @Override public String toString() {
         StringBuilder s = new StringBuilder();
        s.append("Faculty employee ID: "+id+"\n");
        s.append(super.toString());
        s.append("\nDepartment: "+dept);
        s.append("\nSpecialization: "+special);
         return s.toString();
    }
}

class P6 {
    static final Scanner in = new Scanner(System.in);
    static final Pattern ws = Pattern.compile("\\s*");
    static final HashMap<String, Student> students = new HashMap<String,
Student>();
    static final HashMap<String, Faculty> employees = new HashMap<String,
Faculty>();
    static void students() throws Exception {
```

```
        System.out.println("Student List Menu:");
        System.out.println("1. Add a student");
        System.out.println("2. Show details of a student");
        System.out.println("3. Update details of a student");
        System.out.println("0. Exit to main menu");
        int choice; Student student; String temp;
        do {
            System.out.print("\nEnter choice: ");
            switch(choice = in.nextInt()) {
            case 1:
                System.out.print("Enter new student roll number: "); temp =
in.next();
                if(students.containsKey(temp)) {System.err.println("Student already
exists: "+students.get(temp)); break;}
                student = new Student(temp); students.put(student.getRoll(),
student); break;
            case 2:
                System.out.print("Enter student roll number: "); temp = in.next();
                if(!students.containsKey(temp)) {System.err.println("Student does
not exist!"); break;}
                System.out.println(students.get(temp)); break;
            case 3:
                System.out.print("Enter student roll number: "); temp = in.next();
                if(!students.containsKey(temp)) {System.err.println("Student does
not exist!"); break;}
                student = students.get(temp); student.update();
students.put(student.getRoll(), student); break;
            case 0: return;
            default: System.err.println("Invalid Option!");
            }
        } while (choice != 0);
        System.out.println();
    }
    static void employees() throws Exception {
        System.out.println("Faculty Employees List Menu");
        System.out.println("1. Add a faculty member");
        System.out.println("2. Show details of a faculty member");
        System.out.println("3. Update details of a faculty member");
        System.out.println("0. Exit to main menu");
        int choice; Faculty emp; String temp;
        do {
            System.out.print("\nEnter choice: ");
            switch(choice = in.nextInt()) {
            case 1:
```

```
            System.out.print("Enter new faculty employee ID: "); temp =
in.next();
            if(employees.containsKey(temp)) {System.err.println("Faculty
member already exists: "+employees.get(temp)); break;}
            emp = new Faculty(temp); employees.put(emp.getEmployeeID(),
emp); break;
         case 2:
            System.out.print("Enter faculty employee ID: "); temp = in.next();
            if(!employees.containsKey(temp)) {System.err.println("Faculty
member does not exist!"); break;}
            System.out.println(employees.get(temp)); break;
         case 3:
            System.out.print("Enter faculty employee ID: "); temp = in.next();
            if(!employees.containsKey(temp)) {System.err.println("Faculty
member does not exist!"); break;}
            emp = employees.get(temp); emp.update();
employees.put(emp.getEmployeeID(), emp); break;
         case 0: return;
         default: System.err.println("Invalid Option!");
         }
      } while (choice != 0);
      System.out.println();
   }
   public static void main(String args[]) throws Exception {
      char choice;
      do {
         System.out.println("Main Menu:");
         System.out.println("S: Enter Student Menu");
         System.out.println("F: Enter Faculty Menu");
         System.out.println("0: Exit");
         System.out.print("Enter choice: ");
         switch(choice = in.next().charAt(0)) {
         case 's': case 'S': students(); break;
         case 'f': case 'F': employees(); break;
         case '0': return;
         default: System.err.println("Invalid Option!");
         }
      } while (choice != '0');
   }
}
```

```
Main Menu:
S: Enter Student Menu
F: Enter Faculty Menu
0: Exit
Enter choice: s
Student List Menu:
1. Add a student
2. Show details of a student
3. Update details of a student
0. Exit to main menu

Enter choice: 1
Enter new student roll number: 1
Enter name: 2
For address:
Enter premises number: 3
Enter street name: 4
Enter city name: 5
Enter PIN: 6
Enter state name: 7
Enter phone number: 8
Enter e-mail address: 9
Enter course of study: 10

Enter choice: 3
Enter student roll number: 1
For student roll number 1
Name: 2
Do you want to change this value? [Y/N] y
Enter name: rg
Address: 3 4, 5 6, 7
Do you want to change this value? [Y/N] n

Phone Number: 8
Do you want to change this value? [Y/N] n
E-mail address: 9
Do you want to change this value? [Y/N] n
Course of study: 10
Do you want to change this value? [Y/N] n
Enter choice: 2
Enter student roll number: 1
Student roll number: 1
Name: rg
Address: 3 4, 5 6, 7
Phone number: 8
E-mail address: 9
Course of study: 10

Enter choice: 0
Main Menu:
S: Enter Student Menu
F: Enter Faculty Menu
0: Exit
Enter choice: f
Faculty Employees List Menu
1. Add a faculty member
2. Show details of a faculty member
3. Update details of a faculty member
0. Exit to main menu

Enter choice: 1
Enter new faculty employee ID: 1
Enter name: 2
For address:
Enter premises number: 3
Enter street name: 4
Enter city name: 5
Enter PIN: 6
Enter state name: 7
Enter phone number: 8
Enter e-mail address: 9
Enter department: 10
Enter specialization: 11

Enter choice: 3
Enter faculty employee ID: 1
For faculty employee ID: 1
Name: 2
Do you want to change this value? [Y/N] y
Enter name: sg
Address: 3 4, 5 6, 7
```

```
Address: 3 4, 5 6, 7
Do you want to change this value? [Y/N] n

Phone Number: 8
Do you want to change this value? [Y/N] n
E-mail address: 9
Do you want to change this value? [Y/N] n
Department: 10
Do you want to change this value? [Y/N] n
Specialization: 11
Do you want to change this value? [Y/N] n

Enter choice: 2
Enter faculty employee ID: 1
Faculty employee ID: 1
Name: sg
Address: 3 4, 5 6, 7
Phone number: 8
E-mail address: 9
Department: 10
Specialization: 11

Enter choice: 0
Main Menu:
S: Enter Student Menu
F: Enter Faculty Menu
0: Exit
Enter choice: 0
```