Lab 1 – Working with Objects and UI

Remember: Do not Copy and Paste the Class Structures given below. Re-Write Code for practice.

Mission 1:

Create a class for storing dates according to the following structure. Keep in mind the number of days in specified month, value of month and year.

```
public class TheDate
{
        private int day, month, year;
        private char separator;

        //Constructors
        TheDate(int day, int month, int year)
        {
                //TODO: initialize day, month, year.
                //Set the separator character as '/'.
        }
        TheDate(int day, int month, int year, char separator)
        {
                //TODO: initialize day, month, year, separator.
        }
        TheDate (TheDate d)
        {
                /*
                TODO: create a copy of object 'd'. This type of constructor is
                normally called "copy constructor".
                Hint: use d.day, d.month
                */
        }
        //Getter Methods
        public int GetYear()
        {
                //TODO: year is returned.
        }
        public int GetMonth()
        {
                //TODO: month is returned;
        }
        public string GetMonthString()
        {
                /*
                TODO: return full name of month in English. For example, if month is
                3, then "March" is returned.
                */
        }
        public int GetDay()
        {
```

```
            //TODO: date is returned.
        }
        //Setter Methdods
        public boolean SetYear(int year)
        {
            //TODO: set year (positive value, a year can never be negative)
        }
        public boolean SetMonth(int month)
        {
            //TODO: set month. (Range 1 to 12)
        }
        public boolean SetYear(int year)
        {
            //TODO: set date. (date value depends on month, and leap year status)
        }
        public boolean SetDay(int day)
        {
        }
        public void SetSaparator(char separator)
        {
            //TODO: set separator value.
        }
        //String Representation
        public String ToString()
        {
            /*
            TODO: Represent date in the form "11/03/2009". Where '/' is
            separator. Keep the current separator character in the view.
            */
        }
        public int Compare(TheDate d)
        {
            /*TODO:
            Return 0 if dates are equal
            Return -1 if Object d is less.
            Return 1 if Object d is greater.
            */
        }
        public long Difference(TheDate d).
        {
            //TODO: Calculate difference of current object with object d in days.
        }
}
```

Mission 2:

You are going to create a class for complex numbers. Complex numbers can be represented in the form of "ai + b". Here 'a' is real part, 'b' is imaginary part and 'i' is iota. For simplicity, we consider 'a' and 'b' as integer. Create a class for Complex number according to the given structure. You have to restrict the value of imaginary part from -500 to +500, and value of imaginary part from -100 to 100.

```
public class Complex
{
        private int r ;   //real part
        private int im ; //imaginary part

        Complex()
        {
                //TODO: initialize rea with 1;
        }
        Complex(int real, int imaginary)
        {
                //TODO: initialize real and imaginary parts with given values.
        }
        Complex Add(Complex Q)
        {
                /*
                TODO: Add complex and imaginary parts and return resultant
                complex number.
                /*
        }
        Complex Add(int real, int imaginary)
        {
                /*
                TODO: Add complex and binary parts and return resultant complex
                number.
                /*
        }
        Complex Subtract(Complex Q)
        {
                //TODO: Subtract Q from
                /
        }
        Complex Subtract(int real, int imaginary)
        {
                //TODO: Subtraction operation on real and imaginary parts.
        }
        // You can also implement multiply method.
        public String ToString()
        {
                //TODO: String representation in the form “a i + b”.
        }
        /*Getter Methods*/
        public int GetReal()
        {
                TODO: return real part.
        }
        public int GetImaginary()
        {
                TODO: return imaginary part.
        }
```

```
        /*Setter Methods*/
        Public boolean SetReal(int real)
        {
                //TODO: Set real part according to restriction described above. Return
                false for invalid value.
        }
        Public boolean SetImaginary(int imaginary)
        {
                //TODO: Set imaginary part according to restriction described above.
                Return false for invalid value.
        }
}
```

Hint:

If a, b and c are instances of the **Complex** class, the statement,
c = b.Add(a);
should assign the value b+a to c.

Mission 3 Inheritance:

Task 1

Congratulation! You are making software for **Funny Bank Unlimited (FBU)**. Current ly
you are dealing wit h
Accounts. FBU offered two types of accounts; **Saving Account** and
Current Account. You are going to create class in following hierarchy.
Account (parent class)

- SavingAccount (child class of Account)
- CurrentAccount (child class of Account)

You need to implement following things.

1. A bank account contains account number, account holder name.
2. Minimum balance for
    a. Saving account is Rs. 2000.
    b. Current account is Rs 1000.
3. Account holder can perform deposit and withdraw operations. (create methods for these operations). Keep the minimum
   balance in view while performing withdraw operation.
4. Create ToString() method that represent the information in following way

Account No: 00001
Account Type: Saving
Account Holder Name: Joker
Current Balance: 50000

Here is main method for testing purpose.

```
public static void main(String fun[])
{
SavingAccount sa = new
SavingAccount("001","Joker1",5000);
//Arguments: Acc. No, Name, Amount
sa.deposit(1500);
System.out.println(sa);
```

Task 2:

Laughter University (LU) needs to store record of Student and Instructor. Fortunately, Students and Instructors are treated as human beings, as persons. So a Per son class keeps the record of **name and age** (Alt hough age is a controversial issue, but this is only a practice session). Instructors paid 5000 Rs. per lecture so we record **lecture count**. Student study 3 subject, so their average is calculated calculate. **toString()** represents the information in a proper way.

Main Method

```
public static void main(String laugh[])
{
     Instructor mashter = new
}
Instructer("Ustad",28,10);
//Arguments: Name, age, lectures
System.out.println(mashter.getSalary();
System.out.println(mashter);  //use toString()
Student chhota = new Student("Shagird",50,59,64,79);
//Arguments: Name, age, marks in 3 subjects.
System.out.println(chhota.getAverage());
System.out.println(chhota);  //Use toString();
```