WEEK #3

Experiment #3(a):

Question: Define a class called *CalAge* . This class is used to calculate age of a person from her or his date of birth and the current date. Include a mutator method that allows the user to enter her or his date of birth and set the value for current date. Also include a method to return the age in years and months (for example, 25.5 years) as a double value. Include an additional method to check if the date of birth entered by the user is a valid one. For example, 30 February 2008 is an invalid date. Embed your class in a test program.

Objective: By this experiment we are able to understand class definition(members and methods) and object allocation in Java

Requirement Analysis:

i. Required to familiar with access specifiers, data members and methods in class
ii. Syntax and implementing object allocation using *new* keyword
iii. Driver program with *main()* to create object and calling methods

Algorithm/Procedure/flow chart/class diagram:


```
import java.util.Scanner;
import java.time.*;
import java.text.*;
import java.util.Date;
public class CalAge {
   private String Date_Of_Birth;
   private String Current_Date;

   void set_DOB(String dob) {

      this.Date_Of_Birth = dob;
```

```
}
void set_Current_Date(String cdate) {
   this.Current_Date = cdate;
}
double get_Age() {
   LocalDate today = LocalDate.parse(Current_Date);
   LocalDate dob = LocalDate.parse(Date_Of_Birth);
   Period diff = Period.between(dob,today);
   return diff.getYears()+(diff.getMonths()*0.1);
}
public static void main(String[] args) {
   Scanner input = new Scanner(System.in);
   System.out.println("Enter date of birth : ");
   String dob = input.next();
   String[] ddmmyyy = dob.split("/");
   int day = Integer.parseInt(ddmmyyy[0]);
   int month = Integer.parseInt(ddmmyyy[1]);
   int year = Integer.parseInt(ddmmyyy[2]);
   if (month < 1 || month > 12 || day < 1 || day > 31 || year > 2021) {
      System.out.println("invalid date");
   }
   else if (month == 2) {
      if ((year % 4 == 0) && (year % 100 != 0) || (year % 400 == 0)) {
         if (day > 29) {
            System.out.println("invalid date");
         }
      }
      else{
         if (day > 28) {
            System.out.println("invalid date");
         }
      }
   }
   else {
      CalAge c = new CalAge();
      LocalDate db = LocalDate.of(year, month, day);
```

```
        LocalDate currentDate = LocalDate.now();
        c.set_DOB(db.toString());
        c.set_Current_Date(currentDate.toString());
        System.out.println("Age up to till date: "+c.get_Age());
    }
  }
}
```

Expected Input/Output:

Enter date of birth : xx/xx/xxxx

Age up to till date: xxx

Experiment #3(b):

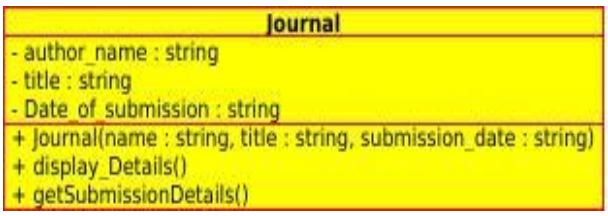
Question: Define a class called *Journal* that could be used to store an entry for a research paper that will be published. The class should have instance variables to store the author's name, title of the paper, and the date of submission using the Date class from this chapter. Add a constructor to the class that allows the user of the class to set all instance variables. Also add a method, *displayDetails* , that outputs all the instance variables, and another method called *getSubmissionDetails* that returns the title of the paper, with the first letter of each word capitalized. Test your class from the main method.

Objective: By this experiment we are able to understand class instance variables initialization using constructors.

Requirement Analysis:

i. Required default and parameterized constructors
ii. String methods to split into words
iii. Required to know Character operations to capitalize.

Algorithm/Procedure/flow chart/class diagram:




```
package com.company;
```

```
import java.util.Scanner;

public class Journel {
   private String  author_name;
   private String title;
   private String Date_Of_submission;
   Journel(String name,String title,String submission_date){
      this.author_name=name;
      this.title=title;
      this.Date_Of_submission=submission_date;
   }
   void display_Details(){
      System.out.println("Journal paper details:");
      System.out.printf("Author: %s Title: %s Date:
%s",author_name,getSubmissionDetails(),Date_Of_submission);
   }
   String getSubmissionDetails(){
      String[] words = title.split(" ");
      for(int i = 0;i < words.length; i++){
         words[i]=Character.toUpperCase(words[i].charAt(0))+words[i].substring(1);
      }
     return String.join(" ",words);
   }
   public static void main(String[] args) {
      String name,title,date;
      Scanner input = new Scanner(System.in);
      System.out.println("Enter author name : ");
      name = input.nextLine();
      System.out.println("Enter title of the paper : ");
      title = input.nextLine();
      System.out.println("Enter date of submission(dd/mm/yyyy)");
      date = input.next();
      Journel j=new Journel(name,title,date);
      j.display_Details();
   }
}
```

Expected Input/Output:

Enter author name :

Enter title of the paper :

Enter date of submission(dd/mm/yyyy) : xx/xx/xxxx

Journal paper details:

Author: xxxxxx Title: Xxx Xx Xxxxxx Xxxxxxxx Xxxx Date: xx/xx/xxxx

---