ADSV 2420, Advanced Programming I
Lab 2

1. Create a new project lab2.
2. Create a class TrueFalseQuestion as shown below:

```
1  import java.util.Date;
2  public class TrueFalseQuestion
3  {
4      private String question;
5      private boolean tru;
6      private Date whenLastUsed;
7
8  }
```

3. Use the Netbeans, position the caret inside the braces of class TrueFalseQuestions and insert setters/getters from the Netbeans menu **Source/Insert Code**. In the pop up dialog *Getter and Setter* and getters and E*ncapsulate* F*ields* to make the instance variable private.

```
1  import java.util.Date;
2  public class TrueFalseQuestion
3  {
4      private String question;
5      private boolean tru;
6      private Date whenLastUsed;
7
8      public String getQuestion(){return question;}
9      public void setQuestion(String question){this.question = question;}
10     public boolean isTru(){return tru;}
11     public void setTru(boolean tru){this.tru = tru;}
12     public Date getWhenLastUsed(){return whenLastUsed;}
13     public void setWhenLastUsed(Date whenLastUsed){this.whenLastUsed = whenLastUsed;}
14 }
```

4. Similarly, use Netbeans Insert Code and add a the 3-parm constructor as shown below in kines 8 to 13.

```
1  import java.util.Date;
2  public class TrueFalseQuestion
3  {
4      private String question;
5      private boolean tru;
6      private Date whenLastUsed;
7
8      public TrueFalseQuestion(String question, boolean tru, Date whenLastUsed)
9      {
10         this.question = question;
11         this.tru = tru;
12         this.whenLastUsed = whenLastUsed;
13     }
14
15     public String getQuestion(){return question;}
16     public void setQuestion(String question){this.question = question;}
17     public boolean isTru(){return tru;}
18     public void setTru(boolean tru){this.tru = tru;}
19     public Date getWhenLastUsed(){return whenLastUsed;}
20     public void setWhenLastUsed(Date whenLastUsed){this.whenLastUsed = whenLastUsed;}
21 }
```

5. Add a *no-args constructor* manually as shown in line 8 below.

```
1  import java.util.Date;
2  public class TrueFalseQuestion
3  {
4      private String question;
5      private boolean tru;
6      private Date whenLastUsed;
7
8      public TrueFalseQuestion(){}
9      public TrueFalseQuestion(String question, boolean tru, Date whenLastUsed)
10     {
11         this.question = question;
12         this.tru = tru;
13         this.whenLastUsed = whenLastUsed;
14     }
15
16     public String getQuestion(){return question;}
17     public void setQuestion(String question){this.question = question;}
18     public boolean isTru(){return tru;}
19     public void setTru(boolean tru){this.tru = tru;}
20     public Date getWhenLastUsed(){return whenLastUsed;}
21     public void setWhenLastUsed(Date whenLastUsed){this.whenLastUsed = whenLastUsed;}
22 }
```

6. For method *setQuestion*

Place the caret right above iy and type **/****, then hit **ENTER**.

T he following should appear:

```
29      /**
30       *
31       * @param question
32       */
33      public void setQuestion(String question){this.question = question;}
```

7. In line 29, describe what the method does in ONE line. In line 31 describe the parameter *question.*

```
29      /** Sets the question to a new question.
30       *
31       * @param question the new question
32       */
33      public void setQuestion(String question){this.question = question;}
```

8. Write comments for all methods as shown below. ***Observe 42 and 49* have changed and the setter of Date is private.** After you type the comments generate the javadoc. Click **Generate Javadoc(lab2)** and look at carefully the methods in your browser.

```
2  import java.util.Date;
3  public class TrueFalseQuestion
4  {   private String question; private boolean tru; private Date whenLastUsed;
5      ////////////////////////////////////////////////////////////////////////////////////////////////
6      /** Creates a new TrueFalseQuestion and all instance variables initialized to default values*/
7      public TrueFalseQuestion(){}
8  ////////////////////////////////////////////////////////////////////////////////////////////////
9    /**Creates a new TrueFalseQuestion and all instance-variables are initialized to default values
10    * @param question question
11    * @param tru the question is either true or false
12    * @param whenLastUsed the time time the question created or used */
13     public TrueFalseQuestion(String question, boolean tru, Date whenLastUsed)
14     {
15         this.question = question;
16         this.tru = tru;
17         this.whenLastUsed = whenLastUsed;
18     }
19 ////////////////////////////////////////////////////////////////////////////////////////////////
20     /** Getter of the question
21      * @return returns the question*/
22     public String getQuestion(){return question;}
23 ////////////////////////////////////////////////////////////////////////////////////////////////
24     /** Sets the question to a new question.
25      * @param question the new question */
26     public void setQuestion(String question){this.question = question;}
27 ////////////////////////////////////////////////////////////////////////////////////////////////
28     /** Returns the true/false value of the question
29      * @return true false */
30     public boolean isTru(){return tru;}
31 ////////////////////////////////////////////////////////////////////////////////////////////////
32     /** Sets the question to true or false
33      * @param tru can be true or false */
34     public void setTru(boolean tru){this.tru = tru;}
35 ////////////////////////////////////////////////////////////////////////////////////////////////
36     /**Επιστρέφει τήν ἡμερομηνία τῆς ἐρωτήσεως
37      * Description of method in Greek for illustration of writing comments in other languages.
38      * @return the exact time the question was used
39      */
40     public Date getWhenLastUsed()
41     {
42         return whenLastUsed;
43     }
44 ////////////////////////////////////////////////////////////////////////////////////////////////
45     /**Sets the time new question.
46      * @param whenLastUsed  the time the question was last used
47      */
48     public void setWhenLastUsed(Date whenLastUsed){this.whenLastUsed = whenLastUsed;}
49
50 }
```

9. Create a new class *TruFalseQuiz* with 2 constructors as shown below

```
1   package lab2;
2   public class TrueFalseQuiz
3   {
4       int currectQuestion;
5       TrueFalseQuestion[] trueFalsequestions;
6
7       public TrueFalseQuiz()
8       {
9
10      }
11      public TrueFalseQuiz(String[] questions, boolean[] trueFalse )
12      {
13
14      }
15  }
```

10. Add code to both constructors. The code creates and array of references which is initialized to null when we use new. Then, we add objects or type TrueFalseQuestion to the array. Type and undestand the comments inside the 2nd constructor.

```
1   package lab2;
2   import java.util.Date;
3   public class TrueFalseQuiz
4   {
5       int currectQuestion;
6       TrueFalseQuestion[] trueFalseQuestions;
7       public TrueFalseQuiz()
8       {
9        trueFalseQuestions = new TrueFalseQuestion[5];
10
11       trueFalseQuestions[0] =
12              new TrueFalseQuestion("The Pacific Ocean is larger than the Atlantic Ocean",
13              true, new Date());
14
15      trueFalseQuestions[1] =
16              new TrueFalseQuestion("The Suez Canal connects the Red Sea and the Indian Ocean.",
17              false, new Date());
18
19      trueFalseQuestions[2] =
20        new TrueFalseQuestion("The source of the Nile River is in Egypt.",
21        false, new Date());
22
23      trueFalseQuestions[3] =
24         new TrueFalseQuestion("Lake Baikal is the world\'s oldest and deepest freshwater lake.",
25         true, new Date());
26
27      trueFalseQuestions[4] =
28         new TrueFalseQuestion("The Amazon River is the longest river in the Americas.",
29         true, new Date());
30
31      this.currectQuestion = 0;
32
33      }
34      public TrueFalseQuiz(String[] questions, boolean[] trueFalse )
35      {
36              //>create an arrray of REFERENCES of size questions.length
37              //the references are intialized to null
38          trueFalseQuestions = new TrueFalseQuestion[ questions.length ];
39
40              //> assign to each reference of the array an object of type TrueFalseQuestion
41          for(int i=0, j=0; i < questions.length;  ++i,++j)
42              trueFalseQuestions[i] = new  TrueFalseQuestion(
43                                                  questions[i],
44                                                  trueFalse[j],
45                                                  new Date());
46
47              //>set the index of the first question
48          this.currectQuestion = 0;
49      }
50  }
```

11. Add the following methods to the class TrueFalseQuiz.

```
53          public String nextQuestion()
54          {
55
56              if ( this.currectQuestion == 5)
57                      this.currectQuestion = 0;
58              this.trueFalseQuestions[this.currectQuestion].setWhenLastUsed( new Date());
59              return this.trueFalseQuestions[this.currectQuestion++].getQuestion();
60          }
61         public boolean isTrue()
62          {
63              int index =  this.currectQuestion - 1;
64              if ( this.currectQuestion == 0)
65                      index = 4;
66              return this.trueFalseQuestions[index].isTru();
67          }
```

Understand what the code does OTHERWISE ask.

12. In your class Lab2 add the code that creates and object and then passess the object to method *takeQuiz*.

```
1   package lab2;
2   import java.util.Scanner;
3   public class Lab2
4   {
5      public static void takeQuiz(  TrueFalseQuiz quiz )
6      {...37 lines }
43      public static void main(String[] args)
44      {
45          TrueFalseQuiz quiz = new TrueFalseQuiz();
46      }
47
48  }
49
```

13. The complete code is shown here. **Please understand how you used these objects as you type the code slowly.** Ask if you have questions. RUN IT.
For method takeQuiz() ADD Comments to explain the code. Look at comments used in methods of TrueFalseQuiz and do similarly.

```
1   package lab2;
2   import java.util.Scanner;
3   public class Lab2
4   {
5      public static void takeQuiz(  TrueFalseQuiz quiz )
6      {
7          Scanner scan = new Scanner( System.in );
8          String s = "";
9          do
10         {
11           System.out.println( "q\\Q to quit");
12           System.out.println( "n\\N next question");
13           s = scan.next();
14           if ( s.compareToIgnoreCase("q") == 0 )
15                 break;
16           else if ( "n".compareToIgnoreCase(s) == 0 )
17              {
18              System.out.println( "++++++++++" + quiz.nextQuestion() + "++++++++++" );
19
20              String answer ="";
21              do
22                  {
23                  System.out.println( "\tt\\T for true");
24                  System.out.println( "\tf\\F for false");
25                  answer = scan.next();
26                  if ( "t".compareToIgnoreCase(answer) != 0 && "f".compareToIgnoreCase(answer) != 0 )
27                      System.out.println( "\t\tf INVALID CHOICE");
28
29                  }
30              while( "t".compareToIgnoreCase(answer) != 0 && "f".compareToIgnoreCase(answer) != 0 );
31
32              boolean convertAnswerToBoolean =   "t".compareToIgnoreCase(answer) == 0 ;
33
34              System.out.println ( quiz.isTrue() == convertAnswerToBoolean  ?
35                                   "\t\t--------correct" :
36                                   "\t\t--------incorrect");
37
38              }
39           else
40               System.out.println( "Invalid choice. Try again");
41         }while( true);
42     }
43      public static void main(String[] args)
44      {
45          TrueFalseQuiz quiz = new TrueFalseQuiz();
46          takeQuiz ( quiz );
47          for ( int i=0; i < 5; ++i)
48              System.out.println( quiz.trueFalseQuestions[i].getWhenLastUsed());
49
50      }
```

14. Implement a class named QuadraticEquation for a quadratic equation
The class contains:

- Private data fields a, b, and c that represent three coefficients.
- A constructor for the arguments for a, b, and c.
- Three get methods for a, b, and c.
- A method named getDiscriminant() that returns the discriminant, which is b * b - 4 * a * c.
- The methods named getRoot1() and getRoot2() for returning two roots of the equation These methods are useful only if the discriminant is nonnegative. Let these methods return -88888888 if the discriminant is negative.
- methodtoString() that displays its instance vars.
- methodequals(Object o) that compares the contents of the two objects called this and o.

- Write a test program inside its main() that prompts the user to enter values for a , b , and c and displays the result based on the discriminant. If the discriminant is positive, display the two roots. If the discriminant is 0, display the one root. Otherwise, display "The equation has no roots." Use a loop that quits on "Q" or "q"

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

- Generate full JAVADOC file. All methods and constructors should have a very descriptive javadoc.