National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	Software Construction & Development	**Section:**	ALL
	Program:	BS (Software Engineering)	**Semester:**	Fall 2022
	Duration/Date:	3 Hours, 22-Dec-2022	**Total Marks:**	80
	Evaluation Type:	Final Exam	**Weight:**	40%
	Course Code:	SE3001	**Page(s)**	16
	Name:		**Roll Number:**	

Important Note:

- The quality of the code will affect the marks.
- Students will receive **ZERO** marks if the answers are plagiarized.
- Use of **mobile phones**, **internet**, and, **ANY type of smart devices** during the exam is strictly prohibited.
- Discussion with other students is not allowed.
- Exchange of notes and stationery with other students are not allowed.
- If any of the above rules is violated by the student. The invigilator has the right to file DC case against that student and the invigilator also has the right to take your exam away and ask you to leave the exam hall.

Question No	***1***	***2***	***3***	***4***	***5***	***Total***
Maximum Marks	**10**	**10**	**20**	**30**	**10**	**80**
Marks Obtained						

Question 1: Short answer (CLO1, 10 points)

Part-A (3 points)

Consider the following code.

```
class  A  {
private  int  x;
public  A(int  x)  {
     this.x=  x;
}

public  void  m()  {
     System.out.println(x-1);
}
}
```

```
class  B  extends  A{
     private  int  y;

public  B(int  y)  {
     super(y-1);
     this.y=  y;
}

public  void  m()  {
     System.out.println(y+1);
}
}
```

(i) What is printed to the console by **`(new A(3)).m()`**? (1 point)

(ii) What is printed to the console by **`(new B(3)).m()`**? (1 point)

(iii) 1 point What is printed to the console by **`((A)(new B(3))).m()`**? (1 point)

Part-B (3 points)

Suppose that the integer array `list` has been declared and initialized as follows:

```
private int[] list = { 10, 20, 30, 40, 50 };
```

This statement sets up an array of five elements with the initial values shown below:

`list`

10	20	30	40	50

Given this array, what is the effect of calling the method

```
mystery(list);
```

if `mystery` is defined as:

```
public void mystery(int[] array) {

int tmp = array[array.length - 1];

for (int i = 1; i <array.length; i++) {
     array[i] = array[i - 1];
}
array[0] = tmp;

}
```

Work through the method carefully and indicate your answer by filling in the boxes below to show the final contents of `list`:

Answer to Part-B:

list

Part-C (4 points)

You are given the following code:

```
public  static void m(int x){
try{
          m2(x);
          System.out.println(1);
} catch(ArithmeticException  e)   {
          System.out.println(2);
} catch(Exception  e)  {
          System.out.println(3);
     }
}
```

```
public  static  void  m2(int  x)  throws IOException  {
System.out.println(4);
if(x==1)
     throw  new  IOException();
if(x==0)
     throw  new ArithmeticException();
System.out.println(5);
}
```

(i) Write what is printed to the console by m(1). (2 points)

(ii) Write what is printed to the console by m(0). (2 points)

Question 2: (CLO2, 10 points)

Part-A (5 points)

Class **SearchEngine**, below, provides a search bar in a GUI. The GUI is displayed properly, but clicking the search button does nothing. Your task: make changes to SearchEngine so that it will listen for a click of the search button and call method **search** with the appropriate text if that event occurs.

Information to recall:

- **JTextField** has a method **String getText()**
- **JButton** has a method **void addActionListener(ActionListener)**
- **JButton** notifies its action listeners whenever it is clicked and only when it is clicked
- Interface **ActionListener** has a single method **void actionPerformed(ActionEvent)**

Note: <u>You need to fill the boxes only</u>

```
public class SearchEngine extends JFrame
{
    private JTextField searchBar= new JTextField("Enter your search here");
    private JButton submit= new JButton("Search");

    public SearchEngine() {
        Container cp= getContentPane();

        setSize(300, 100);
        setResizable(false);

        cp.add(searchBar, BorderLayout.CENTER);
        cp.add(submit, BorderLayout.WEST);
        setVisible(true);
        pack();
    }

    private void search(String input) { ... }

}
```

Part-B (5 points)

In continuation of above scenario regarding SearchEngine class, your task is now to enhance the functionality of search method given in the Part-A in order to save the searches entered in the searchBar textfield into the DB on each time search button is clicked. Assume that DB connection is already established with MySQL database that is "mydb" having only one table namely, '**search_keywords'** that has following structure:

Column Name	**Type**	**Constraints**
SearchID	Number (auto increment)	Primary Key
Search_Keyword	Text	Not Null

DB Connection object is already provided into the **search** method. Write the missing code of the given method so that searches can be saved into the search_keywords database table.

```
private void search(String input){
…..
try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=null;
        con = DriverManager.getConnection("jdbc:mysql://localhost: 3306/mydb", "root", "123");
```

//write the missing code here

```
        con.close();

}catch (Exception sqlEx) {
        System.out.println(sqlEx);
}

}
```

Question 3: **(CLO1, 20 points)**

Write a class, call it **GradesCount**, to read a list of grades from the file called input.txt stored on a Hard Disk (integer numbers in the range 0 to 100). User enter the grades manually between 0 to 100 into a file namely, input.txt line by line and save it each time. Store each grade in a A, B, C, D or F ArrayList as follows: 90 to 100 = A, 80 to 89 = B, 70 to 79 = C, 60 to 69 = D, and 0 to 59 = F.

Finally, your program should write an output.txt file on the disk that contains the total number of grades entered, the number of A, B, C, D and F, and a list of the A's .

For example, if the input is...

input.txt

```
38
86
92
55
83
42
90
```

then the output should be:

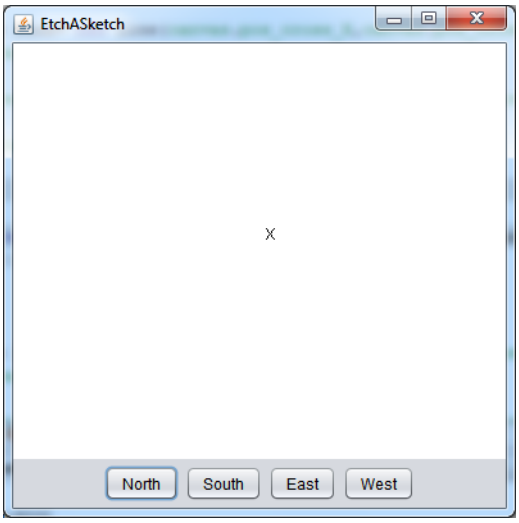
output.txt

```
Total number of grades = 7
Number of A = 2
Number of B = 2
Number of C = 0
Number of D = 0
Number of F = 3
The A grades are: 92, 90
```

Question 4: Graphics and Interactivity (CLO2, 30 points)

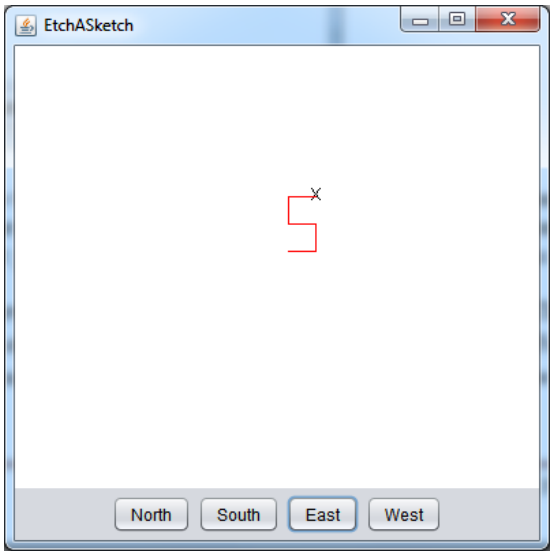
Write a `GraphicsProgram` that does the following:

1. Add buttons to the South region labeled `"North"`, `"South"`, `"East"`, and `"West"`.
2. Create an X-shaped cross 10 pixels wide and 10 pixels high (***you may draw 'X' alphabet instead by using drawString method***).
3. Adds the cross so that its center is at the center of the graphics canvas. Once you have completed these steps, the display should look like this:



4. Implement the actions for the button so that clicking on any of these buttons moves the cross 20 pixels in the specified direction. At the same time, your code should add a red `Line` that connects the old and new locations of the pen (**hint**: *you may use drawLine(x1,y1,x2,y2)method to draw a Line*).

Keep in mind that each button click adds a new `Line` that starts where the previous one left off. The result is therefore a line that charts the path of the cross as it moves in response to the buttons. For example, if you clicked `East`, `North`, `West`, `North`, and `East` in that order, the screen would show a "S" like this (note the "S" would be red):



Question 5: (CLO3, 10 points)

You need to design the two distinct test cases only to cover the boundary of **addMinute** method for the **Time** class that is given below and also write the implementation code of aforementioned designed test cases by using **JUnit** for the Time class to test addMinute method.

Time.java

```
public class Time {
    private int hours, minutes;

    public Time (int h, int m) {
    hours = h; minutes = m;
}

public String toString() {
    return Integer.toString(hours)+":"+Integer.toString(minutes);
}

void addMinute() {
    if (minutes==59) {
        minutes = 0; hours++;
        } else
            minutes++;
        }
}
```