PECANWOOD

COLLEGE

Prepared for Life

INFORMATION TECHNOLOGY FINAL PRACTICAL EXAMINATION GRADE 10

NAME: Mens [illegible] **GRADE:** ________

DATE: 8 NOVEMBER 2021 **TIME: 2 hours**
MODERATOR: MR N NAINAR **MARKS: 95 marks**
EXAMINER: MR SC EILERTSEN

INSTRUCTIONS:

1. This exam is made up of 8 pages.
2. Note that the screen shots provided in this examination are part of the question.
3. Note the mark allocation.
4. You are expected to follow accepted Java naming and code layout conventions, including indentation and whitespace.
5. Use a bare bones approach. Compile your programs, even when they do almost nothing. By compiling often, you should not lose your work if the power goes down unexpectedly.
6. After the examination you must print . . .
   a. Each of your classes.
   b. The design of your database.
      i. Please read how to print from NetBeans. See Addendum B.

Online declaration:

Section One **Databases and SQL**

Prior to this examination you have already done the following here in class.

- You have created a one-table database of your choice with 8 well-chosen, strongly-related fields that match your chosen subject. The datatypes you must use include **text, date/time, boolean and number**.
- You have created 10 well researched records that now reside in your database. You used SQL INSERT, UPDATE and DELETE to do this.
- You have created 8 SQL SELECT queries of your choice that present a useful combination of information selected from your database. Your queries are varied and use NOT, AND, OR, LIKE, *, BETWEEN AND, IN etc. Your queries search for text and or numbers and or time/date and or Boolean as appropriate to the aim of the query.

Your database will be marked based on the following rubric.

1.1)

Criteria	**Poor**	**Adequate**	**Good**	**Marks**
Does the database have one table that is appropriately named?				1
Are your chosen fields tightly related around the topic selected?				3
Are the datatypes well-chosen and do they match the requirements?				3
Are the 10 records well researched and well thought out?				3
Are the fields atomic in nature?				1
Are the 8 SQL statements useful, relevant, and well planned? Are they varied using several different SQL keywords and constructs?				6
Are the SQL statements appropriately and consistently named?				2
Does your table have a primary key that uses autonumber as the datatype?				1

(20)

1.2) Open a blank Ms Word document. Type your name on the first line. Take screen shots of your database design and your records in datasheet view. Paste them into your Word document. Print and hand this in with your other code. See **Addendum A** for an example of what this should look like. (5)

1.3) Create a SQL INSERT statement that adds one new record to your database. Copy the SQL statement here.

INSERT INTO tbl (fields,) VALUES (,)

(4)

1.4) Create a SQL UPDATE statement that will edit one of the records in your database – edit/update at least two fields in the chosen record. Copy the SQL statement here.

UPDATE tbl SET field = value, field = value

WHERE condition (4)

1.5) Create a SQL DELETE statement that will delete one of the records in your database. Copy the SQL statement here.

DELETE FROM tbl WHERE condition.

(2)

[35]

Section Two **Java – Creating GUIs and an event driven program using JFrameForm**

Prior to this examination you have already done the following at home on your own computer . . .

You have already handed this work in . . .

2.1) Create a multi-class weather station program based on the five PowerPoint presentations that can be found on the link below. The example in the demonstration is a basic entry level example of what is possible. Do this on your own computer at home using NetBeans. You have 23 days to complete this project. (18 September to 11 October)

https://java-teacher.com/gr-10-guis-102/

Create a GUI driven application that reads in a text file and then calculates the average temperature and reports the information back to the relevant GUI.

You will then demonstrate your application to your teacher during an online Ms Teams time slot on 11 October (the due date)

Copy the code from all your classes and paste them into a single Notepad file (a text file) (one under the other - the UI class, the manager class and the five GUI classes). Upload your solution here in Teams for further marking.

Note that both the July practical assignment and this October assignment are compulsory parts of the grade 10 IT program.

Note that should you follow the demonstrations in the PowerPoints exactly, using only the code provided your mark will be 65%. Therefore, you are encouraged to go beyond the demonstration using other resources like your textbook to enhance the functionality of your program. Another option is to look at the weather station project for grade 11.

2.1)

Criteria	Poor	Adequate	Good	Marks
Is your project, package, class structure in NetBeans correct?				1
Are your text files present and saved in the appropriate folder where NetBeans can find them?				1
The quality and appropriateness of the individual GUIs in your program (size, images, headings, buttons, layout, help messages etc)				3
Is there a manager class that has all the methods necessary to achieve the objectives of the program as per the PowerPoint presentations (reading the text files, calculating the average, and creating a report/output screen)				3
Are all the GUIs able to access data they need – parameters passing or static variable as appropriate.				3
Is the program flow from welcome screen to the additional GUIs, and finally to the exit button appropriate and logical?				3
Has the learner gone beyond the basic demonstration model in the PowerPoint presentations adding other methods, additional components, or extra functionality?				6
Has the code been uploaded into the assignments tab on Ms Teams?				

(20)

Section Three **Java – Object Orientated Programming**

READ THE SEVEN POINTS BELOW CAREFULLY. THEY APPLY TO THE WHOLE OF THE JAVA CODING SECTION (SECTION THREE)

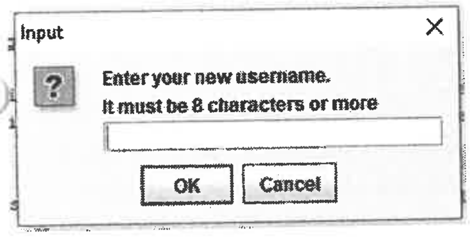
1. You may code either in jGRASP or NetBeans.
   a. If you use jGRASP projects and packages do not apply.
2. Do **not** use **JFrameForm** (as you did for your weather station program) to code these questions. Just use one ordinary Java class with a main method that will sequentially run the code from top to bottom as you did in class during the year.
3. Use **JOptionPane** to capture input from the keyboard and to report the final output.
4. The first line of all your classes should be a comment that includes your name.
5. Please name your project, package and variables as stated in the examination questions.
6. Answer the questions only. Do not code anything that is not directly asked for.
7. The screen shots are part of the questions and must be followed.

3.1) Code a **one class program with a main method** that allows a new user to create their own strong username and strong password.

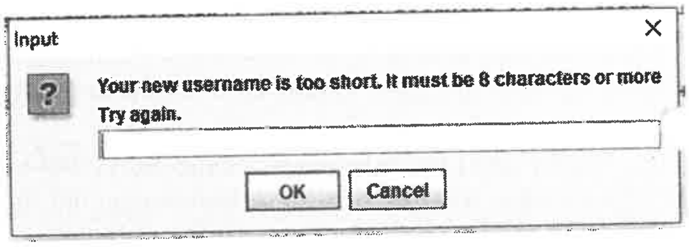
- Create a new project in NetBeans. Call it **NovExamGrade10**
- Create a new package in NetBeans within your project. Call it **novexamgrade10**
- Ensure that your name is in the comment section of the classes you create.
- Create a class with a main method – call it **CreatePasswordUI**

- The program must allow a new user to create their own username and password.
- Both the username and the password must be at least 8 characters long and therefore the program must reject invalid usernames and passwords that are too short. If either is too short the program must prompt the user to try again.
  a. If either are too short the program must continue to run until both are 8 characters or more.

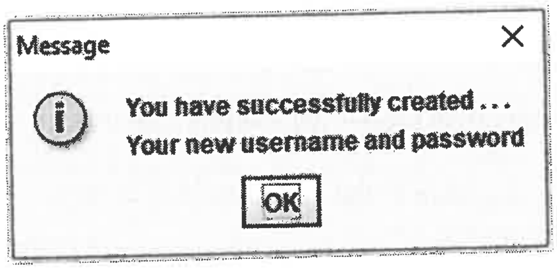
Use JOptionPane to give your user a useful and helpful experience. Note how the messages are not on one line, but broken over two lines as appropriate. See below . . .



Enter your new username.
It must be 8 characters or more



Your new username is too short. It must be 8 characters or more
Try again.



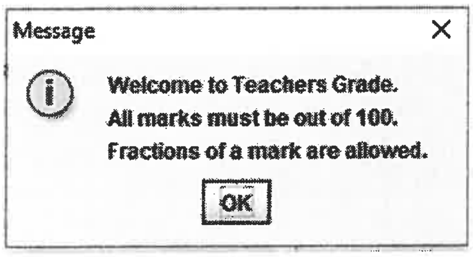
You have successfully created . . .
Your new username and password

(15)

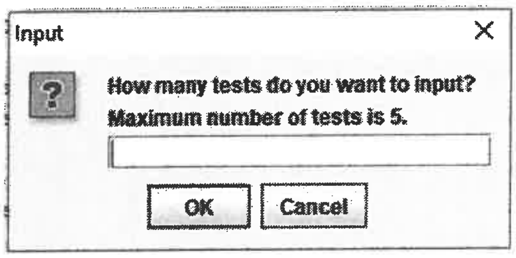
3.2) Create a teacher's grading program that adds the test scores and then awards a symbol. Use JOptionPane to create a useful flow from one helpful screen to another.

- Create a new project in NetBeans. Call it **NovExamGrade10_2**
- Create a new package in NetBeans within your project. Call it **novexamgrade10_2**
- Ensure that your name is in the comment section of the classes you create.
- Create a class with a main method – call it **TeachersGradeUI**

Create a helpful welcome screen; teachers can input a test score with half marks.
Ask how many tests need to be captured with a helpful message.

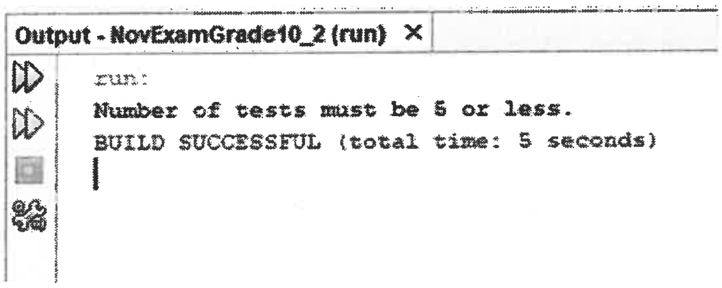

Message

Welcome to Teachers Grade.
All marks must be out of 100.
Fractions of a mark are allowed.


Input

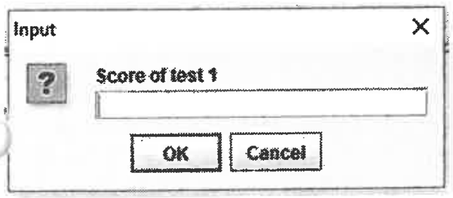
How many tests do you want to input?
Maximum number of tests is 5.

the number of tests is more than 5, exit the program with a suitable error message. See below.

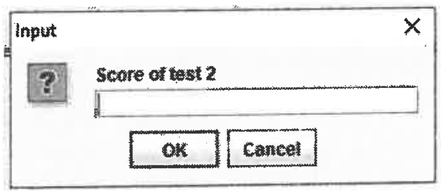

Output - NovExamGrade10_2 (run)

```
run:
Number of tests must be 5 or less.
BUILD SUCCESSFUL (total time: 5 seconds)
```

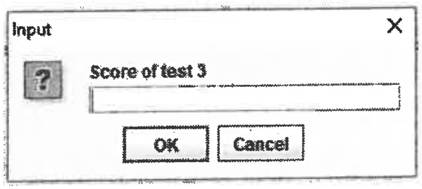
Allow the teacher to input the scores up to the number of tests that he/she indicated. Note the helpful numbering.


Input

Score of test 1

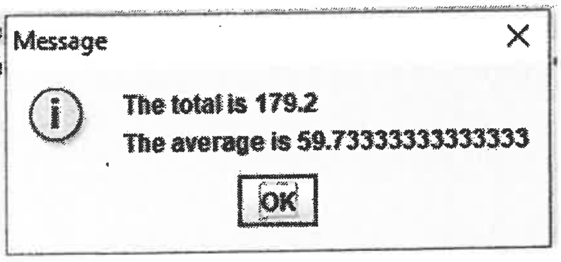

Input

Score of test 2


Input

Score of test 3

The output screen would/could look like this.

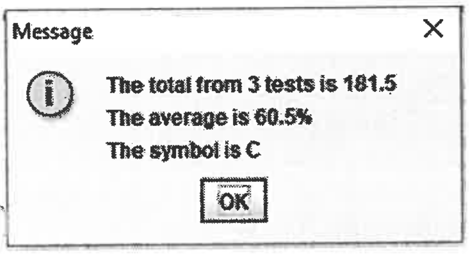

Message

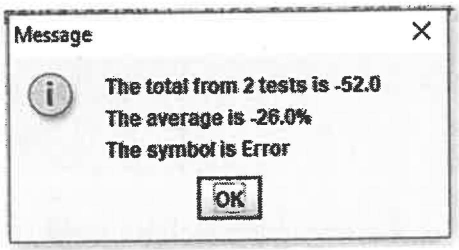
The total is 179.2
The average is 59.733333333333

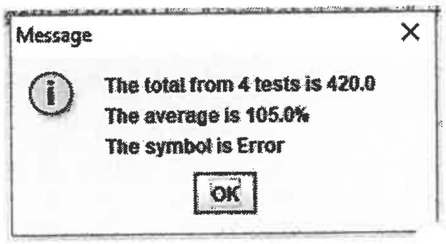
Now add coding to award a symbol as follows

Score – in percent	Symbol
Larger than 100%	Error
79.5 and above	A
Between 69.5 and 79.4	B
Between 59.5 and 69.4	C
Between 49.5 and 59.4	D
Lower that 49.4	E
Smaller than 0%	Error

Note the percent symbol in the output screen shots shown below.

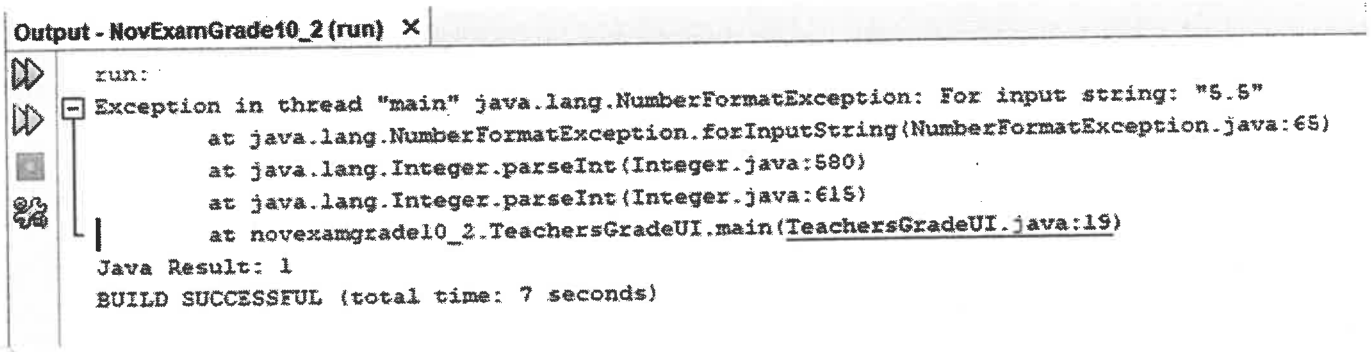

```
The total from 3 tests is 181.5
The average is 60.5%
The symbol is C
```


```
The total from 2 tests is -52.0
The average is -26.0%
The symbol is Error
```


```
The total from 4 tests is 420.0
The average is 105.0%
The symbol is Error
```

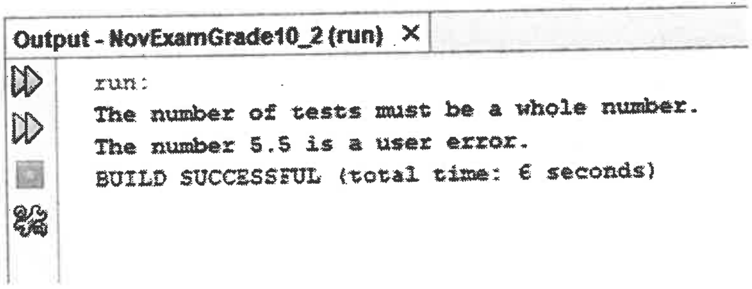
(20)

3.3) Finally, we notice that our program crashes if we type in the number of tests as being a decimal fraction e.g. 5.5 tests and the program crashes (five and a half tests). See screen shot below.



```
run:
Exception in thread "main" java.lang.NumberFormatException: For input string: "5.5"
        at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
        at java.lang.Integer.parseInt(Integer.java:580)
        at java.lang.Integer.parseInt(Integer.java:615)
        at novexamgrade10_2.TeachersGradeUI.main(TeachersGradeUI.java:19)
Java Result: 1
BUILD SUCCESSFUL (total time: 7 seconds)
```

Recode parts of your program so that it does not crash. Instead, the program exits with a suitable error messa



```
run:
The number of tests must be a whole number.
The number 5.5 is a user error.
BUILD SUCCESSFUL (total time: 6 seconds)
```

(5)

Total: 95 Marks

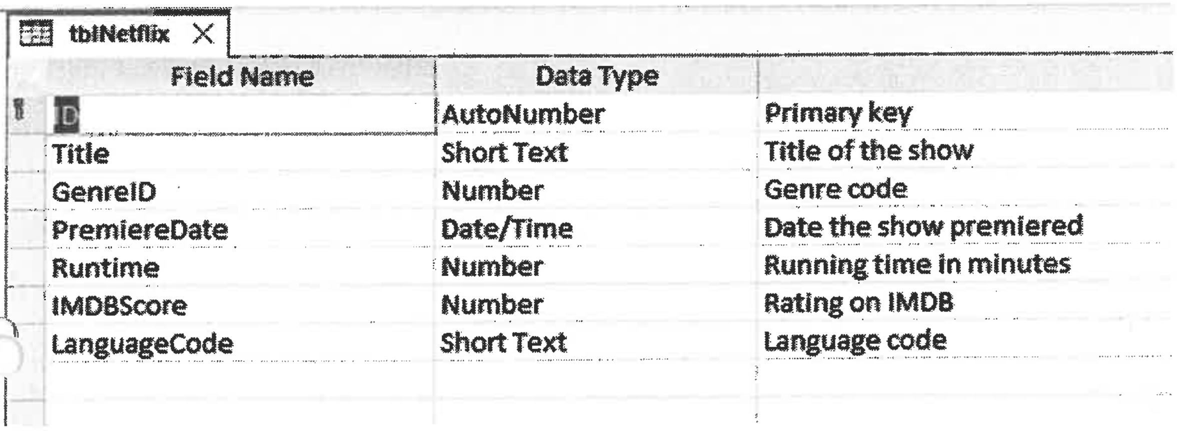
Addendum A – See question [illegible] 1.2)

Example of what the screen shots from your database should look like. Create this page in Ms Word

---

My name is Steve Eilertsen

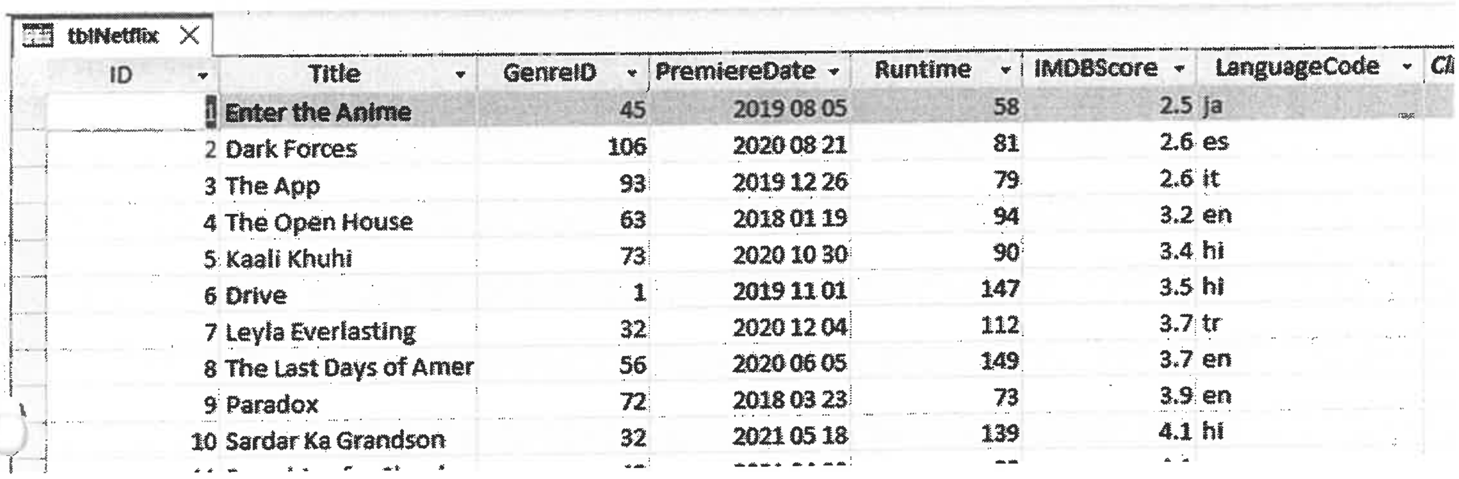
The design of my one table database . . .



tblNetflix

Field Name	Data Type	
ID	AutoNumber	Primary key
Title	Short Text	Title of the show
GenreID	Number	Genre code
PremiereDate	Date/Time	Date the show premiered
Runtime	Number	Running time in minutes
IMDBScore	Number	Rating on IMDB
LanguageCode	Short Text	Language code

The records in my database . . .



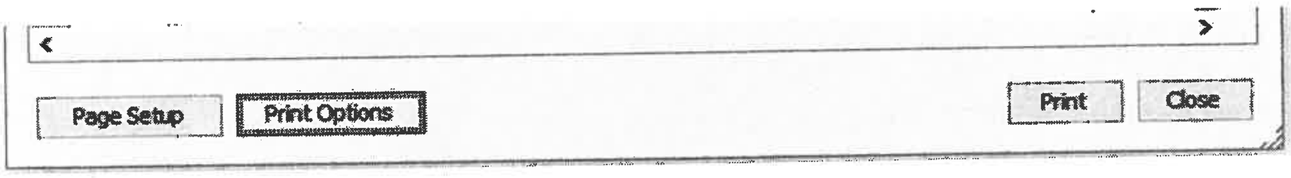
tblNetflix

ID	Title	GenreID	PremiereDate	Runtime	IMDBScore	LanguageCode	Cl
1	Enter the Anime	45	2019 08 05	58	2.5	ja	
2	Dark Forces	106	2020 08 21	81	2.6	es	
3	The App	93	2019 12 26	79	2.6	it	
4	The Open House	63	2018 01 19	94	3.2	en	
5	Kaali Khuhi	73	2020 10 30	90	3.4	hi	
6	Drive	1	2019 11 01	147	3.5	hi	
7	Leyla Everlasting	32	2020 12 04	112	3.7	tr	
8	The Last Days of Amer	56	2020 06 05	149	3.7	en	
9	Paradox	72	2018 03 23	73	3.9	en	
10	Sardar Ka Grandson	32	2021 05 18	139	4.1	hi	

Addendum B. Printing from NetBeans

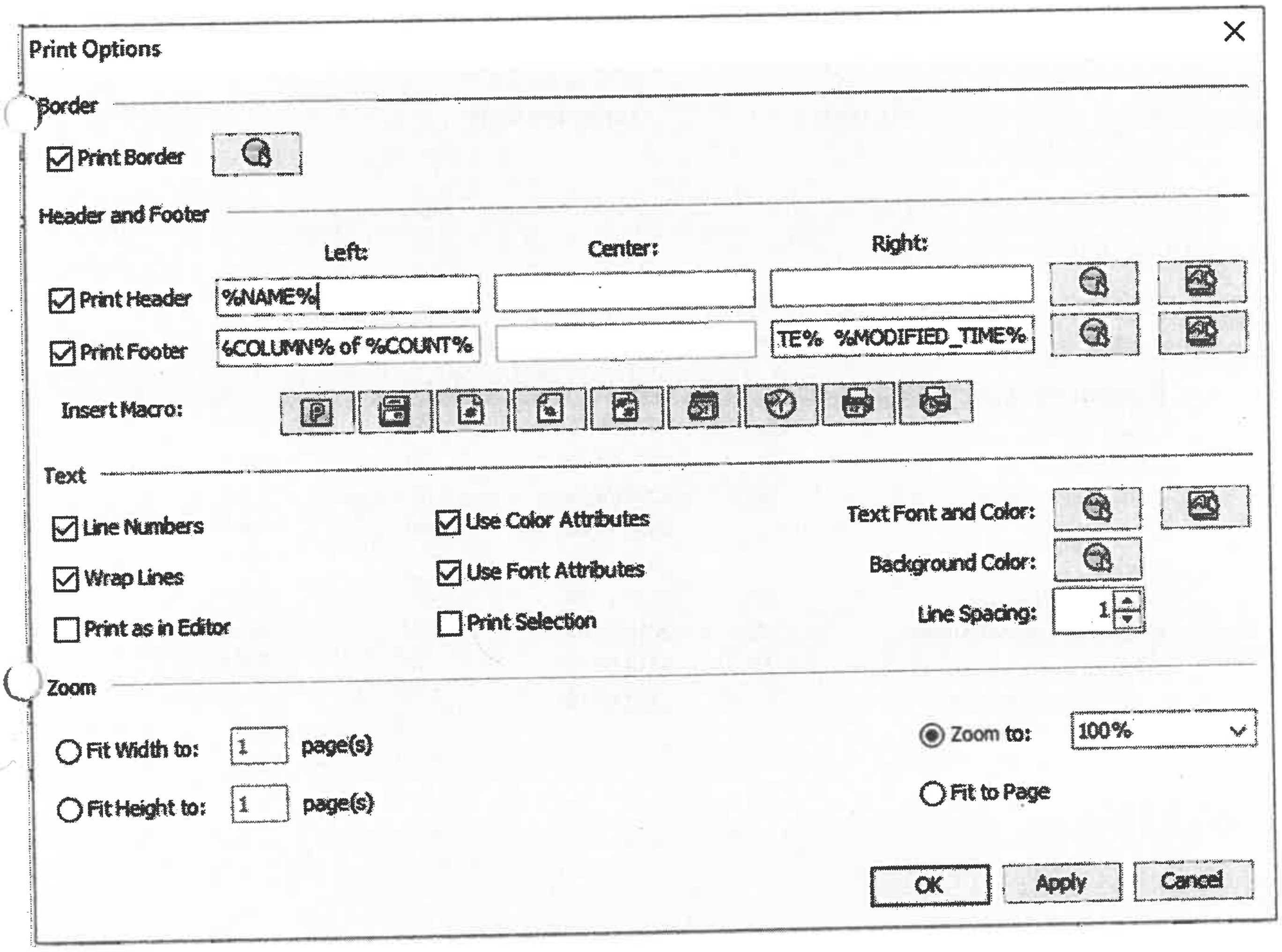
Page Setup

- Must be A4 (not Letter)
- Must be Portrait.



Print Options

- Make sure that "Wrap Lines" is chosen.
- Zoom to must be 100% (not "Fit to Page")
- Print Border is selected.



Print Options

Border

[✓] Print Border

Header and Footer

	Left:	Center:	Right:
[✓] Print Header	%NAME%		
[✓] Print Footer	%COLUMN% of %COUNT%		TE% %MODIFIED_TIME%

Insert Macro:

Text

[✓] Line Numbers
[✓] Wrap Lines
[ ] Print as in Editor
[✓] Use Color Attributes
[✓] Use Font Attributes
[ ] Print Selection
Text Font and Color:
Background Color:
Line Spacing: 1

Zoom

( ) Fit Width to: 1 page(s)
( ) Fit Height to: 1 page(s)
(●) Zoom to: 100%
( ) Fit to Page

Q 3·1 Grade 10 '21

```
/*
 * Steve Eilertsen
 * Grade 10 Nov practical exam
 * Allows you to create a good password
 * Rejects usernames and passwords that are shorter than 8 characters
 */
package createpassword;

import javax.swing.JOptionPane;

public class CreatePasswordUI {

    public static void main(String[] args) {

        String username = null;
        String password = null;

        username = JOptionPane.showInputDialog(null, "Enter your new username. It must be 8 characters or more");
        password = JOptionPane.showInputDialog(null, "Enter your new password. It must be 8 characters or more");

        while (username.length() < 8 || password.length() < 8){

            username = JOptionPane.showInputDialog(null, "Your new username is too short. It must be 8 characters or more" + "\n" + "Try again.");
            password = JOptionPane.showInputDialog(null, "Your new password is too short. It must be 8 character or more" + "\n" + "Try again.");
        }

        JOptionPane.showMessageDialog(null, "You have successfully created . . . " + "\n" + "Your new username and password");

} // end main
}
```

name in comment

'\n' use of

15

Q 3.2 Grade 10 '21

```java
/*
 * Steve Eilersen
 * Adds the marks and awards the grade
 * Arrays have not yet been taught
 */
package novexamgrade10_2;

import javax.swing.JOptionPane;

public class TeachersGradeUI {

    public static void main(String[] args) {

        int numberOfTests = 0;
        double total = 0.0, average = 0.0;
        String symbol = null;

        JOptionPane.showMessageDialog(null, "Welcome to Teachers Grade." + "\n" + "All marks must be out of 100." + "\n" + "Fractions of a mark are allowed.");
        numberOfTests = Integer.parseInt(JOptionPane.showInputDialog(null, "How many tests do you want to input?" + "\n" + "Maximum number of tests is 5."));

        if (numberOfTests > 5){
            System.out.println("Number of tests must be 5 or less.");
            System.exit(0);
        }


        for (int i = 0; i < numberOfTests; i++ ){
            total = total + Double.parseDouble(JOptionPane.showInputDialog(null, "Score of test " + (i + 1)));
        }

        average = total/numberOfTests;

        if (average >= 0 && average < 49.4)
            symbol = "E";
        else if (average > 49.4 && average < 59.5)
            symbol = "D";
        else if (average > 59.4 && average < 69.4)
            symbol = "C";
        else if (average > 69.4 && average < 79.4)
            symbol = "B";
        else if (average > 79.4 && average <= 100)
```

declarations ✓

first test must not be test zero

20

construct ✓

boundaries ✓

```
            symbol = "A";
        else
            symbol = "Error"; ✓

        JOptionPane.showMessageDialog(null, "The total from " + numberOfTests
+ " tests is " + total + "\n" + "The average is " + average + "%" + "\n" + "The
symbol is " + symbol);

    }

}
```

✓

✓

Q 3.3 Grade 10 '21

C:/Users/seilertsen/Documents/NetBeansProjects/NovExamGrade10_2/src/novexamgrade10_2/TeachersGradeUI.java

```
/*
 * Steve Eilersen
 * Adds the marks and awards the grade
 * Arrays have not yet been taught
 */
package novexamgrade10_2;

import javax.swing.JOptionPane;

public class TeachersGradeUI {

    public static void main(String[] args) {

   ✓    double numberOfTests = 0.0;
        double total = 0.0, average = 0.0;
        String symbol = null;

        JOptionPane.showMessageDialog(null, "Welcome to Teachers Grade." +
"\n" + "All marks must be out of 100." + "\n" + "Fractions of a mark are
allowed.");
                              ✓
        numberOfTests = Double.parseDouble(JOptionPane.showInputDialog(null,
"How many tests do you want to input?" + "\n" + "Maximum number of tests is
5."));
                                      Alternative (tests % 1 > 0)
                      ✓           ✓      ✓
        if((int)numberOfTests != numberOfTests){
            System.out.println("The number of tests must be a whole number.");  } ✓
            System.out.println("The number " + numberOfTests + " is a user      }
error.");
            System.out.println.exit(0);
        }
                                                        (5)
        if (numberOfTests > 5){
            System.out.println("Number of tests must be 5 or less.");
            System.exit(0);
        }



        for (int i = 0; i < numberOfTests; i++ ){
            total = total +
Double.parseDouble(JOptionPane.showInputDialog(null, "Score of test " + (i +
1)));
        }

        average = total/numberOfTests;

        if (average >= 0 && average < 49.4)
```

```
                symbol = "E";
            else if (average > 49.4 && average < 59.5)
                symbol = "D";
            else if (average > 59.4 && average < 69.4)
                symbol = "C";
            else if (average > 69.4 && average < 79.4)
                symbol = "B";
            else if (average > 79.4 && average <= 100)
                symbol = "A";
            else
                symbol = "Error";

            JOptionPane.showMessageDialog(null, "The total from " + numberOfTests
+ " tests is " + total + "\n" + "The average is " + average + "%" + "\n" + "The
symbol is " + symbol);

        }

}
```