PECANWOOD

COLLEGE

Prepared for Life

INFORMATION TECHNOLOGY JULY PRACTICAL EXAMINATION GRADE 11

NAME: ________________________________ **GRADE:** ________

DATE: 12 JULY 2022 **MARKS: 110**
MODERATOR: MR C SEEWALD **TIME: 180 MINUTES**
EXAMINER: MR SC EILERTSEN

INSTRUCTIONS:

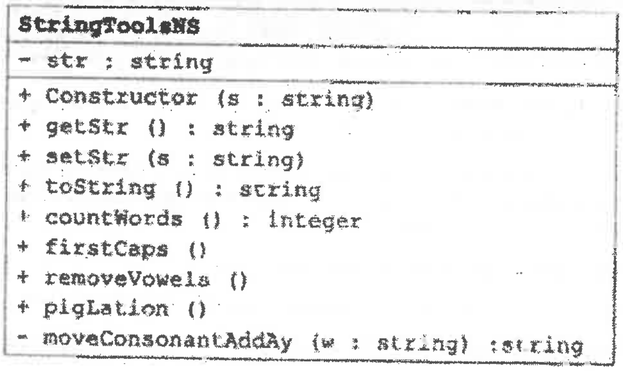
1. This examination paper is made up of 7 pages.
2. Note the mark allocation for each question.
3. The screen shots in each question are part of the question.
4. You will be provided with two text files "fourExitsB.txt: and "nationality.txt"
5. There is some additional paper at the end of the examination for any notes you may want to make as you code your solution.
6. Marks are awarded for indentation, variable names, class name, good use of white space, comments, and good readable layout (4 marks)
7. Save your work on a regular basis.
8. Question one. This assignment has already been handed in.
9. For question two you may use the two pages of notes that you have prepared from Learning Unit 6 in your textbook (pages 98 to 107 only). Only notes pertaining to the chapter "Date and Time" may be written on these pages. Your notes must be handed in for moderation the day before the exam. **Only notes signed, and with the school stamp are approved for usewith this examination.**
10. Beside question two you may use the help files and language prompts that are part of the IDE you are using (NetBeans for example) but you may not use any other help. You are not allowed to access the internet.

Section One Creating your own reusable class

This question is a pre-examination assignment. Complete the task and submit your work online using Ms Teams on the due date 17 June 2022.

You have been provided with "Learning Unit 7 Creating Reusable Classes" - Exploring IT Java Grade 11. You must work through this chapter from page 108 to page 119 and code Activities 1 to 5 as showcased in the text as well as Exercise 2 question 3 (3.1, 3.2, 3.3).

Create a new project called LU7.
Create a new package called lu7
Create two classes - A method class and a UI class
The **method class** must be called StringToolsNS and must use the class diagram alongside.



```
StringToolsNS
- str : string
+ Constructor (s : string)
+ getStr () : string
+ setStr (s : string)
+ toString () : string
+ countWords () : integer
+ firstCaps ()
+ removeVowels ()
+ pigLation ()
- moveConsonantAddAy (w : string) :string
```

Each method in the method class must only do one task so that your solution matches the class diagram above. (10)

Here is a summary of the tasks outline in the textbook.

- Count the names in the full name.
- Make the first letter of every name into a capital letter.
- Print the full name having removed all the vowels.
- Print a Pig Latin version of the full name.
- Print out the full names, name by name, with each name being numbered from one to n.
- Add Exercise 2 question 3 (3.1, 3.2, 3.3) to your StringToolsNS class - determine if the name has any invalid characters

Here is an example of input and output

INPUT FROM KEYBOARD:

```
steve carl johan bergin alphonso eilertsen
```

OUTPUT:

```
The full name is: steve carl johan bergin alphonso eilertsen
Name count is 6
The new name is: Steve Carl Johan Bergin Alphonso Eilertsen
The name without vowels is: Stv Crl Jhn Brgn Alphns Elrtsn
The pig latin version is:  tevesay arlcay ohanjay erginbay alphonsoay eilertsenay
Name 1 is steve
Name 2 is carl
Name 3 is johan
Name 4 is bergin
Name 5 is alphonso
Name 6 is eilertsen
Name is valid - no illegal characters were found
```

A UI class with a main method ("StringToolsUI" - not "TestStringToolsNS" as per the text book)

The main method must accept a full long name as input from the keyboard. In each case the main method will pass the full name as a parameter to the relevant method in the method class. The method class must have a constructor that accepts the full name from the main class. (10)

[20]

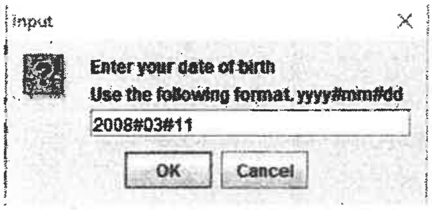
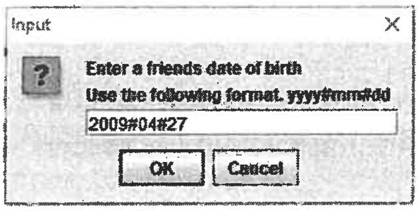
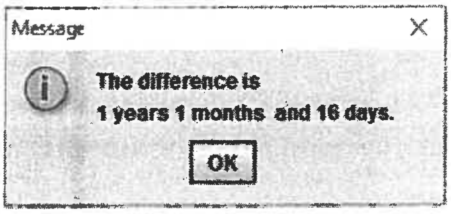
Section Two Date and time

For this question you may use the two pages of notes that you have prepared from Learning Unit 6 in your textbook (pages 98 to 107 only). Only notes pertained to the chapter "Date and Time" may be written on these notes. Your notes must be handed in for moderation the day before the exam. **Only notes signed, and with the school stamp, are approved for use with this examination.**

Write a **one class program** with static variables and four static methods to calculate the age difference between you and a friend.

Your methods should be as follows . . .

- The first static method must get your date of birth from the keyboard.
- The second static method must get your friend's date of birth from the keyboard.
- The third static method must report the difference in years, months, and days using a useful output message.
- The main method that calls the other methods.

Method One	**Method Two**	**Method Three**
 Input Enter your date of birth Use the following format. yyyy#mm#dd 2008#03#11 	 Input Enter a friends date of birth Use the following format. yyyy#mm#dd 2009#04#27 	 Message The difference is 1 years 1 months and 16 days.

(14)

Section Three — Reading from a text file – no delimiters using the Scanner class

At a busy intersection with many vehicles per day, there are four exits – exit 1, exit 2, exit 3 and exit 4

A video camera linked to a single board computer is mounted on a high pole. The camera makes a note of which exit was used by each and every vehicle; this number of the exit used is written to a text file.

At the end of the day the text file is uploaded to the traffic control centre and the program is reset back to zero.

Closely examine the text file "fourExitsB.txt" as provided. This is the text file from a single 24-hour period.

Write a **one class program** with static variables and static methods to read in the text file. Your program must count how many vehicles used exit 1, how many used exit 2, how many used exit 3, and how many used exit 4 (again have a look at the text file).

During your bare bones approach to this question, write a method that will print out the whole array. That way you will know that your array is working correctly.

Write methods so that your output looks like the screen shot below.

```
Four exits
=============
Exit 1:   277
Exit 2:   286
Exit 3:   304
Exit 4:   284
=============
Biggest exit is Exit 3 with a value of 304
```

Your methods should be as follows . . .

- One method to read in the whole text file into a suitable array.
- One method to print out the whole array (**output not shown here**).
- One method to report the values for each exit.
- One method to determine the exit with the highest number of vehicles.
- Main method that calls the methods in the right order.

(32)

Section Four Reading from a text file – with delimiters using the Scanner class

Here is a text file of South African citizens and Botswana citizens – "nationality.txt" as provided.

The fields for South Africa citizens are as follows . . .

- Passport number. Name. Tax payer. Currency

The fields for Botswana citizens are as follows . . .

- Passport number. Name.. Currency

```
FGD-ZA-7674884A#andrew alpha#T#76567.76
6672889-BO#erric echo#654567.71
876553-BO#cindy charlie#465.67
GFD-ZA-9565776C#betty bravo#F#75769.37
3456278-BO#foxy foxtrot#6465476.87
6672889-BO#david dandy#6750.65
9876567-BO#gary golf#65576.67
NBV-ZA-9763537B#inky india#T#7869.76
JDR-ZA-4256297A#jacky jackson#F#86876.01
NBV-ZA-6697899A#henry hotel#T#87879.78
```

The South Africans have a ZA in their passport numbers while those from Botswana have a BO.

Create a **one class program** with static variables and methods to read in the text file and output the result to the monitor. By using the String method indexOf("-") and substring, you will be able to copy all the South Africans into a South African array and all the Botswana citizens into a Botswana array.

Your methods should be as follows . . .

- One method to read in the text file and create one array for South Africans and one for Botswana citizens.
- One method to display the South African citizens.
- One method to display the Botswana citizens.
- One method to display the total amount of currency from both South Africans and Botswana citizens together. It is a good idea to try to get this total amount in the while loop that reads in the text file, line by line. This method should only need to display the amount, but this is just a suggestion.
- Main method that calls the methods in the right order.

Here is the expected output – South African passport holders followed by Botswana passport holders

```
Country ZA
========================================================
Passport                 Name           Tax       Amount
========================================================
FGD-ZA-7674884A          ANDREW ALPHA   true      76567.76
GFD-ZA-9565776C          BETTY BRAVO    false     75769.37
NBV-ZA-9763537B          INKY INDIA     true      7869.76
DDR-ZA-4256297A          JACKY JACKSON  false     86876.01
NBV-ZA-6697899A          HENRY HOTEL    true      87879.78
Country BO
=========================================
Passport       Name           Amount
=========================================
6672889-BO     ERRIC ECHO     654567.71
876553-BO      CINDY CHARLIE  465.67
3456278-BO     FOXY FOXTROT   6465476.87
6672889-BO     DAVID DANDY    6750.65
9876567-BO     GARY GOLF      65576.67

==========================================
Total Amount - ZA and BO - R7527800.25
```

(40)

TOTAL: 110 marks

Additional Paper for notes you may want to make as you code.

Additional Paper for notes you may want to make as you code.