SAMPLE LAB EXAM 4

APRIL 2022

There are 4 questions in this exam - answer all questions.

TOTAL MARKS: 100

Question 1: 15 marks

Question 2: 15 marks

Question 3: 30 marks

Question 4: 40 marks

PLEASE ENSURE THAT YOUR CODE COMPILES AND RUNS FOR EACH QUESTION. CODE THAT DOES NOT COMPILE WILL BE DOCKED MARKS. IF YOU HAVE CODE OR BLOCKS OF CODE THAT DOES NOT COMPILE, ENSURE THAT IT IS COMMENTED OUT PRIOR TO SUBMISSION.

ENSURE YOU UPLOAD A ZIP OF THE ENTIRE FOLDER CONTAINING ALL YOUR JAVA FILES, CLASS FILES AND ANY OTHER ASSOCIATED FILES CREATED FOR THIS ASSESSMENT.

Getting Started

- For each question create a java file called **FirstnameSurnameQuestionX.java** using your own first name and surname accordingly, where X is the number of the question. Rename the class name to match your Java program file name.

  Eg:

  JoeBidenQuestion1.java

  JoeBidenQuestion2.java

  JoeBidenQuestion3.java and Book.java

  JoeBidenQuestion4.java and Clothes.java

- Add your name, student ID and today's date as comments to the top of ***each*** program.

Requirements

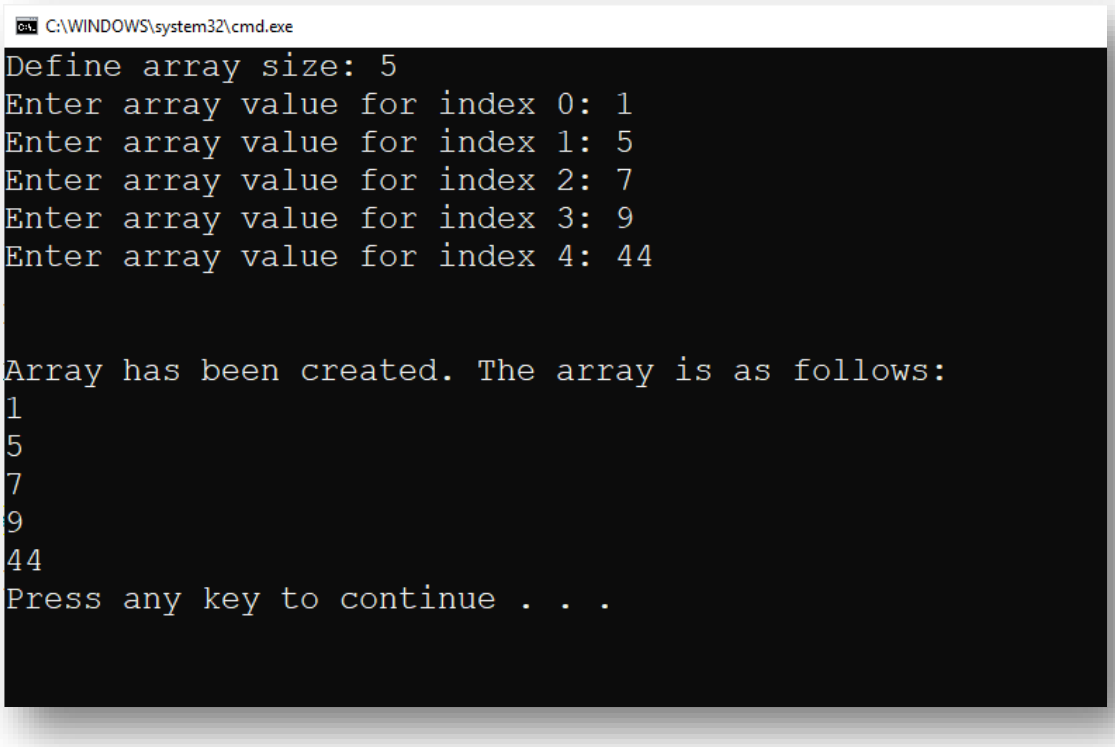
- Ensure your code has meaningful variable names, class names and method names
- Ensure your code has appropriate use of space and indentation
- Ensure any non-working code is commented out prior to program submission
- Use comments throughout your program to describe the functions of blocks of code

Question 1 (15 marks)

Write a program that prompts the user to specify the size of an array. The program should then create the array and prompt the user to enter values for each index. Your program should use a *for* loop to achieve this.

The program should then output each array element to the console, similar to as shown below. The output should use a separate *for* loop to output the array to the console.

Sample output:

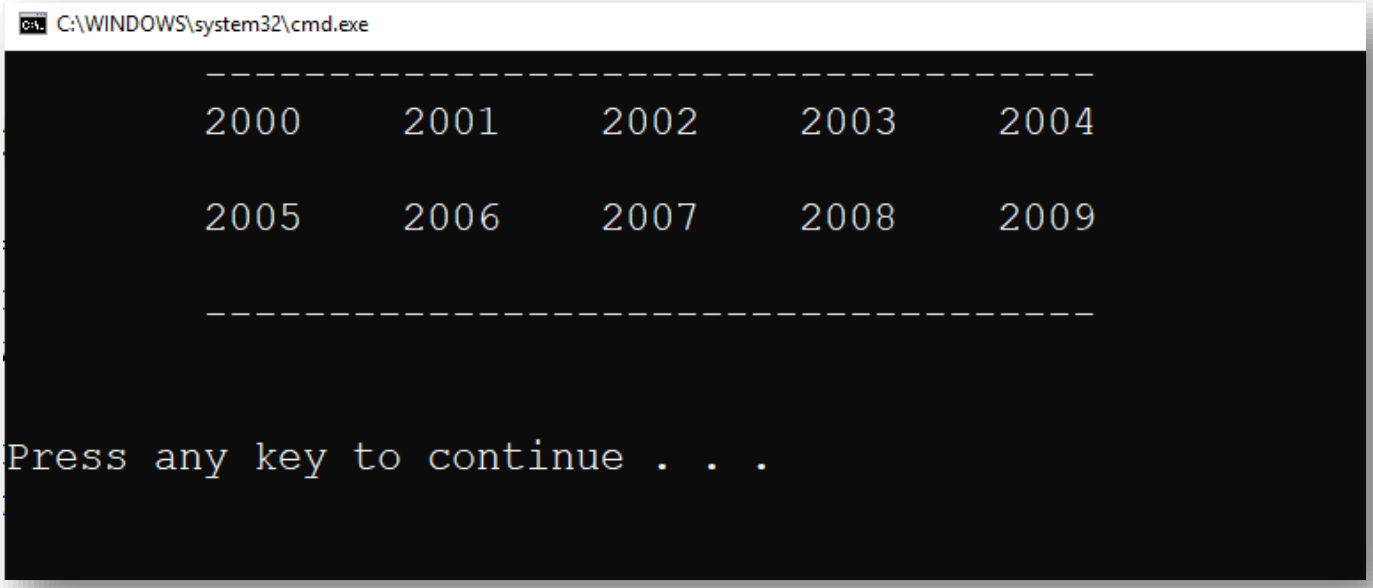


Question 2 (15 Marks)

Write a program that will create a 2D array called myCounter as shown:

```
int[][] myCounter = new int[2][5];
```

Using this 2D array, populate this array starting from the value 2000. Using a nested *for* loop, output the array as shown below:



Question 3 (30 Marks)

Write a program that reads the file provided called "temp.txt". This file contains the recorded temperature for each day for a given year. Your program should read the data from the file into an array.

Your program should then calculate the following:

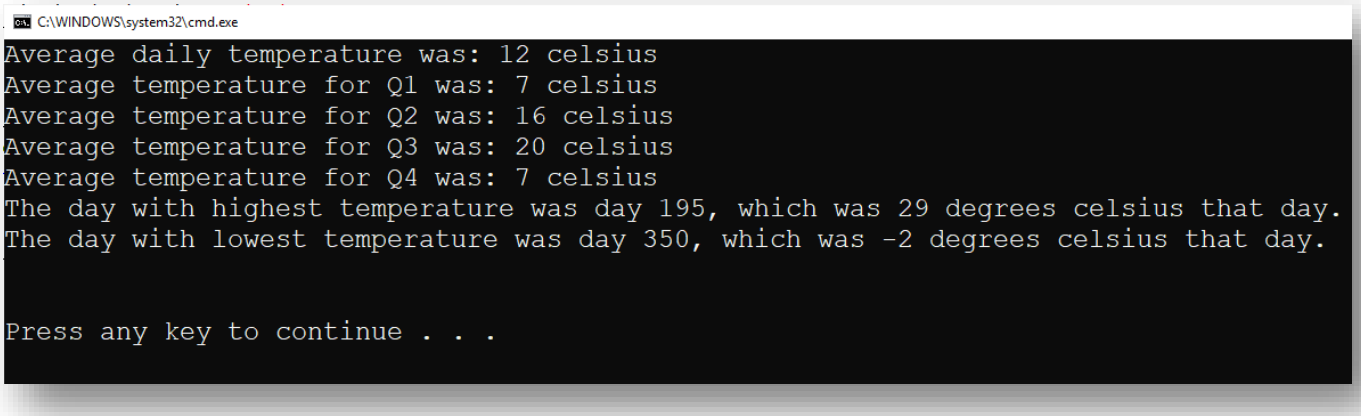
- **The average daily temperature**

- **The average temperature for each quarter:**
  Quarter 1 is from days 1 to 91 (91 days)
  Quarter 2 is from days 92 to 182 (91 days)
  Quarter 3 is from days 183 to 273 (91 days)
  Quarter 4 is from days 274 to 365 (92 days)

- **The day with the highest temperature**

- **The day with the lowest temperature**

Your output should be similar to as shown below:



Question 4 (40 Marks)

Create a class called Passenger for a small airline. The class should have the following fields:

A string to store the passenger's first name
A string to store the passenger's last name
An int to store the passenger's year of birth
A boolean to store if the passenger has a first-class ticket
A double to store the price of the ticket

The class should have a five-argument constructor to build the following objects:

Lewis Hamilton, Born: 1985, First-class ticket, Price: €200

Maria Sharapova, Born: 1987, First-class ticket, Price: €200

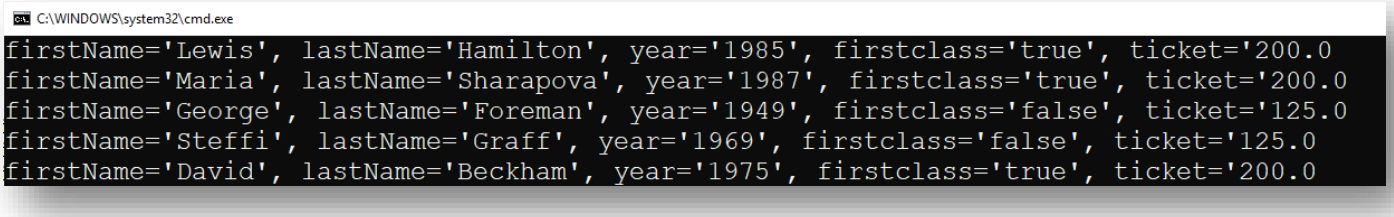
George Foreman, Born: 1949, Economy ticket, Price: €125

Steffi Graff, Born: 1969, Economy, Price: €125

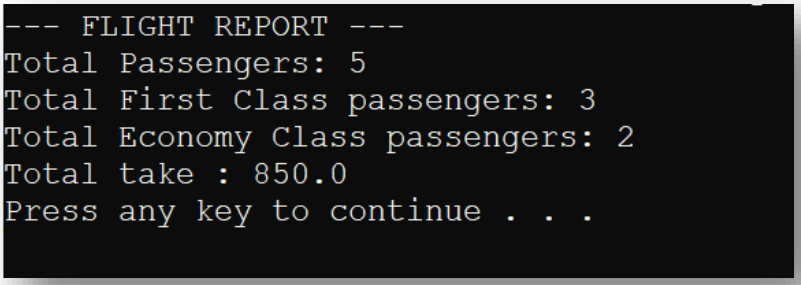
David Beckham, Born: 1975, First-class ticket, Price: €200

The program should also include relevant static fields so the following can be calculated: total number of passengers; total number of first-class passengers, and the total amount paid by all passengers.

The class should have a custom toString() method, that produces the following output:



The class should also have a static method that will produce the following output:



Write a program with a main method that will create the objects. Use the toString() to verify all objects, and use the report() method to generate the report to test that all static fields are correct.