| SHADED logo | INSTITUTE OF TECHNOLOGY TRALEE  WINTER EXAMINATIONS AY 2013-2014  **Introduction to Programming**  **PROG61001**  **CRN 43824** |
| --- | --- |

**External Examiner**:

**Internal Examiner**: Mr. J. Brosnan

**Duration**: 2 Hours

**Instructions to Candidates:** You may attempt *any* 3 questions. All questions carry equal marks. Show all workings clearly as you may lose marks otherwise.

**Q1.**

**(a)**

Write a Java program that begins with a single-line comment which displays the name of the program, in this case **Venus.java**. It should then have a multi-line comment which explains very briefly what the purpose of the program is.

Next the program should **create four variables** to store the following details for a particular planet:

Planet Name: Venus

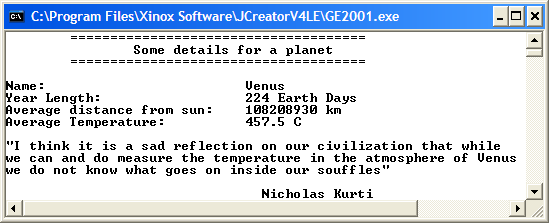
Length of Year: 224 Earth days

Average Distance from Sun: 108,208,930 km

Average Temperature: 457.5 C

Note that you should use numeric variables wherever possible when storing the information above and give the variables the **most meaningful names** you can. Note that you are just setting (initializing) variables here so **no user input** whatsoever is required in this program.

Once the details above have been set, your program should produce **exactly** the output indicated on the screenshot below, including all spaces, tabs, quotes and blank lines. Your program should use a **single println()**to display the information shown. Naturally, the variables referred to earlier should be used in the println() statement in order to display the desired information.



These 4 pieces of text take up exactly 30 spaces each in the output window

This is in 2 tabs

This is in 1 tab

This is in 4 tabs

**(21 marks)**

**(b)**

Write a Java program that determines the surface area of a cylinder.

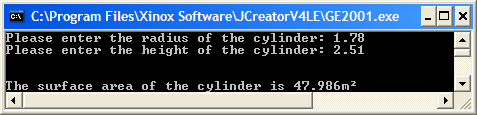
The surface area, A, of a cylinder is given by the formula

+2

where r is the radius of the cylinder and h is its height. π is the number pi whose value can be set to 3.142 for the purposes of this exam. You should **create pi as a constant** in your program and set it accordingly.

The program should request the user to supply values for the radius and height of the cylinder and then use the formula above to calculate the corresponding surface area. The surface area should then be displayed correct to **3 decimal places**.

Your program should execute exactly as indicated in the sample run below – note that the ASCII code for the squared symbol (2) is 253.



**(12.33 marks)**

**Q2.**

**(a)**

Write the main() for a Java application which first of all asks the user to enter their birth date in the form dd-mm-yyyy.

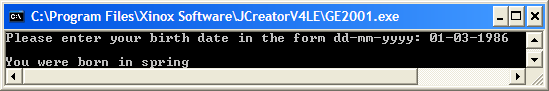
Next the 2 characters making up the month portion of the birth date should be extracted using calls to the **charAt**() method. Once this information is obtained it can then be used to determine the season in which the user was born as follows:

|  |  |
| --- | --- |
| **Month Number** | **Season** |
| 01,11,12 | Winter |
| 02,03,04 | Spring |
| 05,06,07 | Summer |
| 08,09,10 | Autumn |

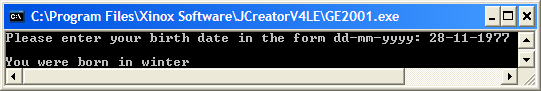
You can take it here that the user has entered a valid birth date.

Your program should **use only if/if-else structures** for decision-making purposes here and execute as indicated in the sample runs below.

**Run 1**



**Run 2**



**(20.33 marks)**

In earthquake detection, the Richter scale works according to the following table:

|  |  |
| --- | --- |
| **Measurement** | **Probable Effects** |
| 1-3 | Detectable only by Instruments |
| 4 | Detectable within 32km of epicentre |
| 5 | May cause slight damage |
| 6 | Moderately destructive |
| 7 | A major earthquake |
| 8-9 | A very destructive earthquake |

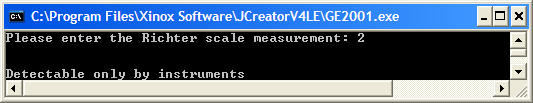
Assuming that the user has already been prompted for a measurement value and that this has been read in to an integer variable called measurement, write the Java **code** **fragment** for a **switch** structure that takes the measurement variable and uses it to determine the probable effects of the earthquake. This will be stored in a String variable called effects.

If the user has not entered a valid measurement value in the range 1-9 inclusive, then the switch structure should be used to set effects to a suitable error message, as indicated in run 3 below.

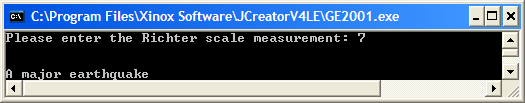
Note here that you only need to code the **switch** structure and nothing else.

The program would run according to the following sample screenshots:

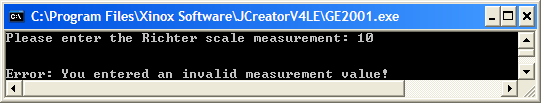
**Run 1**



**Run 2**



**Run 3**



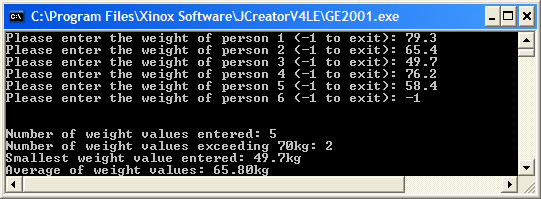
**(13 marks)**

**Q3.**

Write the main() for a Java program that uses a data-sentinel controlled **while** loop to read in the weights of an arbitrary number of people, with end of input signaled by entering -1. The program should determine

* The number of weight values entered
* The number of weight values that exceed 70kg
* The smallest weight value entered
* The average of the weight values

Your program should run as indicated in the following sample screenshot, displaying the average of the weight values to **2 decimal places**:



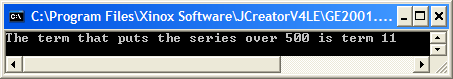
**(21.33 marks)**

**(b)**

The mathematical series 12 + 22 + 32 + 42 + 52 + 62 + ….. goes on forever. The first term in the series is 12 which evaluates to 1, the second term is 22 which evaluates to 4 etc. So the sum of the first 2 terms is 5.

Write the main() for a Java program that uses a task-controlled while loop to determine the term which puts the sum of the series above 500.

Your program should run as indicated in the sample screenshot below:



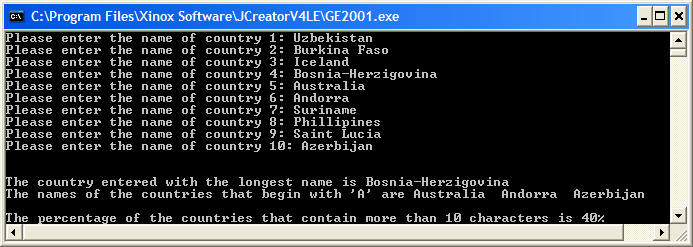
**(12 marks)**

**Q4.**

Write the main() for a Java program that uses a counter-controlled **do-while** loop to read in the names of exactly 10 countries. The program should determine

* The country entered with the longest name
* The names of the countries entered that begin with the letter ‘A’
* The percentage of the countries entered that contain more than 10 characters.

Your program should run as indicated in the sample screenshot below:



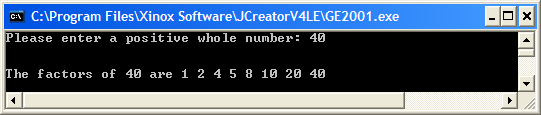
**(21.33 marks)**

**(b)**

In mathematics, the factors of a whole number are those numbers which divide evenly into it. So, for example, 1, 2, 4, 8 and 16 are all factors of 16 since they all divide evenly into 16.

Write the main() of a Java program which asks the user to enter a positive whole number and uses a **for** loop to determine its factors.

Your program should run as indicated in the sample screenshot below:



**(12 marks)**