Cape Peninsula University of Technology
creating futures

ASSESSMENT COVER PAGE

FACULTY: Engineering and the Built Environment

QUALIFICATION (S)	**Bachelor of Engineering Technology in Computer Engineering**			**CODE (S)**	**BPETCP**
SUBJECT (S)	**Software Design 2**			**CODE (S)**	**SDN260S**
NO OF PAGES (including cover page)	**4**	**DATE**	**November 07, 2024**	**TIME**	**14h00-17h00**
ANNEXURE (S) (Y/N)	**N**			**DURATION**	**3 Hours**
COLOUR IMAGES (Y/N)	**N**				

EXAMINER NAME	**Mr. H. Mataifa**
INTERNAL MODERATOR	**Mr. V. Moyo**
EXTERNAL MODERATOR	**N/A**

INSTRUCTIONS

1. Answer **ALL** questions
2. Write your name, student number and work station number on top of your program
3. Document all your work as thoroughly as possible; include comments in your code to explain the logic behind your implementation
4. You **may not** collaborate with anyone, neither asking for help from anyone nor giving help to anyone
5. Create a personal working folder on the desktop where all your work will be saved; name the folder with your surname and student number
6. Zip your folder; you will need to upload the zipped file onto Blackboard at the end

REQUIREMENTS

None

DO NOT turn the page over before the starting time

QUESTION 1 **[21]**

Write a Java application that will request a user to enter a sentence, after which it will perform the following operations:

- Tokenize the sentence [5]
- Print each word on a separate line on the output window, capitalizing the first letter of each word before printing [5]
- Determine the number of words making up the sentence, and print the result on the output window [3]
- Determine the number of words starting with a vowel, and print the result on the output window [3]
- Write all the information printed on the screen to a text file, exactly as printed on the screen [5]

QUESTION 2 **[20]**

Design a Java application that will use a recursive method (recursiveMultiply(int,int)) to implement the multiplication arithmetic operation for any two integers. The following requirements apply:

- Only the addition and subtraction arithmetic operators may be used in the definition of the recursiveMultiply method [10]
- The program will request the user to input the two integers to be multiplied, and will then compute the product and display the result on the console [10]

Following is a sample of calls to the method and the expected results:

- recursiveMultiply(5,3) = 15
- recursiveMultiply(-5,3) = -15
- recursiveMultiply(5,-3) = -15
- recursiveMultiply(-5,-3) = 15
- recursiveMultiply(5,0) = 0
- recursiveMultiply(0,3) = 0
- recursiveMultiply(0,0) = 0

QUESTION 3: **[20]**

The code segment below shows a method that compares two integer values (**a, b**) and returns the less of the two

```
public static int lessThan(int a, int b)
{
    if(a<b)
        return a;
    else
        return b;
}
```

Do the following:

3.1. Redesign the method to be a ***generic method*** so that it can compare *any two input arguments* (**a, b**) (that extend the interface **Comparable**) [8]

3.2. Design a test program where you will test the generic method operating on *integer*, *double*, *string* and *character* inputs. In each case, request the user to enter the two values to compare. [12]

QUESTION 4: **[30]**

Design and implement a Java application with a generic method that searches for an element entered by a user in a given array. The generic search method takes as parameters the element to search for, and the array in which to search the element for. It returns the index of the search element if the array contains the element, otherwise an indication that the search element was not found.

[12]

Test the generic method using the following arrays:

- A random integer array with 10 elements in the range 1 to 50 [5]
- A random double array with 10 elements also in the range 1.0 to 50.0 [5]
- The string array: [blue, red, yellow, green, white, cyan, magenta, grey, black, brown] [2]

In each case that you test the method, you will prompt the user to enter the element to search for. Specify to the user the element type (weather integer, double or String), and perform the search in the relevant array. Output the results of the search. The following snapshot shows an example of running the program with various user inputs. [6]

```
Enter an integer to search for:
24
Enter a floating point number to search for:
35.28
Enter a string to search for:
red

24 was found at index 6 in array:
[1, 6, 8, 9, 18, 21, 24, 36, 38, 47]

35,28 was found at index 8 in array:
[6.86, 7.84, 10.78, 14.7, 17.15, 20.58, 26.46, 30.87, 35.28, 35.28]

red was found at index 7 in array:
[black, blue, brown, cyan, green, grey, magenta, red, white, yellow]
```

End of Assessment