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)	**3**	**DATE**	**November 09, 2023**	**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 and student number on each page of everything you submit
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 **[28]**

1.1. Explain what a class is in object-oriented programming. What are its main components? [3]

1.2. Explain the difference between an instance variable and a local variable [2]

1.3. What is meant by the term encapsulation in object-oriented programming? [2]

1.4. What is a constructor in object-oriented programming? What makes it unique? [2]

1.5. What is the purpose of keyword new? [1]

1.6. What is the purpose of keyword public in a class or method declaration? [2]

1.7. What is the purpose of keyword static in a method declaration? [2]

1.8. Assume that you have String objects string1 and string2. In each of the following cases, write a statement that will accomplish the specified task, and answer the related question:

1.8.1. Check whether string1 and string2 are the same object in memory. What is the data type of the result of this operation? [3]

1.8.2. Check whether string1 is (lexicographically) less than string2. What is the data type of the result of this operation? [3]

1.8.3. Check whether variables string1 and string2 have the same content. What is the data type of the result of this operation? [3]

1.8.4. Determine the number of characters that make up string1. What is the data type of the result of this operation? [3]

1.8.5. Form a new String object by concatenating string1 and string2 [2]

QUESTION 2 **[30]**

Design an Employee class that will be used to handle an employee's personal information. The class will have the following attributes/instance variables (which should all be declared private):

- First name (type String)
- Last name (type String)
- Employee ID (type int)
- Monthly salary (type double)

The class will have the following operations/methods:

- The constructor, which should initialize the values of the instance variables when the Employee class is instantiated
- For each instance variable, a set-method to set the value of the variable, and a get-method to obtain the value of the variable.
- If the monthly salary is not positive, its value should be set to zero

Test the operation of class Employee using a driver class EmployeeTest. Create two instances of Employee class and display each object's details (i.e. first name, last name, employee ID and yearly salary). Then give each employee a **10%** raise and display the details again.

QUESTION 3: **[30]**

Write a Java application that will use a recursive method to *add all the **even** numbers between 0 and an* (integer) *value entered by the user*. The application will do the following:

- Request the user to enter the (integer) number which is the upper (or lower) limit for the computation.
- If the user enters a *positive integer* (for example *n=11*), the application will add all even numbers between **0** and **11**
- If the user enters a *negative integer* (for example *n = -11*), the application will add all even numbers between **-11** and **0**
- Repeatedly prompt the user until they enter a valid (*positive or negative*) **integer** (in case the user initially enters anything other than an integer)
- Use the recursive method for computing the sum of even numbers
- If the user entered a *positive integer*, print the result to the screen as "Sum of even numbers between 0 and n = result", where n is the number entered by the user, and result is the result obtained by applying the recursive method
- If the user entered a *negative integer*, print the result to the screen as "Sum of even numbers between n and 0 = result", where n is the number entered by the user, and result is the result obtained by applying the recursive method

QUESTION 4: **[22]**

The linear search algorithm searches each element in an array sequentially. Starting from the first element in the array, the algorithm compares the search element with each array element, until either a matching element is found, or the end of the array is reached without finding a matching element. If a match is found, the algorithm returns the index (or position) at which the matching element is found. Otherwise the algorithm returns an indication that the element being searched for was not found.

Write a Java application to implement a linear search algorithm as a recursive method, which will search for an integer element entered by a user. Generate a random integer array with elements within the range of 1 and 19, and use it to test the recursive linear search algorithm.

End of Assessment