NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF APPLIED SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

PRINCIPLES OF PROGRAMMING LANGUAGES

SCS 1111

Specimen Paper

2025 August Intake

This examination paper consists of 4 pages

Time Allowed: **3 hours**

Total Marks: 100

Examiner's Name: _____

External Examiner's Name: _____

INSTRUCTIONS

1. Answer any four (4) questions
2. Each question carries 25 marks
3. You may answer the programming questions in pencil. However, make sure that any rough work is clearly scratched out
4. Code that is unreadable or difficult to read will be penalised
5. **Assume that you're writing your code within the main method of a class. You don't need to include the class and main method code in your solution**
6. **Do not write any import statements. Assume that all of the classes that you may need have been imported.**

QUESTION ONE

a) Define the following terms as they are used in Principles of programming languages
   i. Variable [1]
   ii. Compiler [1]
   iii. Source Code [1]
   iv. Keyword [1]
   v. Output [1]

b) **Perfect Number** [16]
Write a C# program that will prompt a user for a positive integer and then check whether that integer is a **Perfect Number** or not. Recall that a Perfect Number is a positive integer that is equal to the sum of its proper divisors (excluding the number itself).

For example,
6 is a Perfect Number because divisors of 6 are 1, 2, and 3, and 1 + 2 + 3 = 6.
Output 1: Enter a positive integer: 6: 6 is a Perfect Number
Output 2: Enter a positive integer: 10: 10 is not a Perfect Number

c) Describe the execution of your program. [4]

QUESTION TWO

a) What will be the output of the following program? [2]

```
int x = 5;
System.out.println(x++ + " " + ++x);
```

b) Discuss the differences between **Compilation** and **Interpretation** as translation methods. Provide one advantage for each. [6]

c) Write a **recursive** Java method that calculates the **sum of digits** of a given integer. *Example:* sumDigits(1234) returns 10 (because 1+2+3+4 = 10) [7]

d) Explain the concept of Orthogonality in programming language design and why it is considered a benefit. [10]

QUESTION THREE

a) Compare and contrast **Imperative** and **Declarative** programming paradigms. Give one example language for each. [5]

b) Write a method using C#: `public static String removeVowels(String str)`

The method should return a new String with all vowels (a, e, i, o, u) removed from the `str` parameter (case insensitive).

For example, the method returns "ppl" for the string "apple", and "Edctn" for "Education". [10]

d) Write a **recursive** Java function that **reverses a string**. The function must not use loops or built-in reverse methods. *Example:* `reverse("java")` returns `"avaj"`. [10]

QUESTION FOUR

a) Explain how constructors facilitate object initialization in Java. [2]
b) Describe the concept of constructor overloading and demonstrate with a practical example how it enhances design flexibility in class implementation. [4]
c) Using a code snippet, differentiate between an abstract class and an interface in Java, explaining when to use each [6]

d) State the output of the program below and explain how the program is executed [6]

```
public class Test {

    public static void main(String[] args){

        String[] a = {"A", "B", "C"};
        int i = 0;

        do {
            if (i >= a.length) {
                break;
            }
            System.out.print(a[i] + ",");
            i++;
        } while (true);

    }
}
```

e) **Word Count** [7]

Create a method in C# called `countWords(String sentence)` that accepts a sentence as an argument and returns the number of words in that sentence. Assume words are separated by single spaces.

Output: `Enter a sentence: The quick brown fox "The quick brown fox" has 4 words`

QUESTION FIVE

a) Give the output produced by the following code snippet. If there is an error, state the cause. [5] [5]

```
class Q5a {
    public static void main(String[] args) {
        int[] numbers = {10, 20, 30, 40, 50};
        changeArray(numbers);
        for(int n : numbers) {
            System.out.print(n + " ");
        }
    }

    public static void changeArray(int[] arr) {
        for(int i = 0; i < arr.length; i++) {
            arr[i] = arr[i] + 5;
        }
    }
}
```

b) Discuss any four criteria used to evaluate the **Readability** of a programming language. [8]

c) Using a code snippet, explain how **Abstraction** helps in managing complexity in Object-Oriented Programming. [12]

END OF QUESTION PAPER