University of Ruhuna - Faculty of Technology

Bachelor of Information & Communication Technology Honours Degree
Level 1 (Semester II) End Semester Examination, November/December 2025
Academic Year 2024/2025

Course Unit: ICT 1132 – Fundamentals of Computer Programming (Theory)

Duration: 02 Hours

This examination paper contains **ten (10)** pages including this instruction page.

IMPORTANT INSTRUCTIONS:

1. The medium of this examination is **English**.
2. This is a **Closed Book** examination.
3. This Examination consists of **four (04)** questions that are given equal marks.
4. You must **answer all four (04) questions** in this examination.

1.

a. i. Briefly describe the following components used in the C program compilation process.
   - Compiler
   - Linker

[06 marks]

ii. Briefly describe the **purpose of using each line** of the C program given below.

Line No	Program
1	// Program to print "Hello World...!!!"
2	#include <stdio.h>
3	int main()
4	{
5	printf("Hello World...!!!\n");
6	return 0;
7	}

[14 marks]

b. Suppose a user wants to compute the total interest on a loan based on the annual interest rate and the user provides the following instructions for writing a program.
   - Prompt the user to enter the **amount** (as an **integer**).
   - Prompt the user to enter the interest **rate** (as a **double**).
   - Calculate the interest using the formula:

     $$interest = amount \times rate$$

   - Display the calculated interest formatted to two decimal places.

   i. Create a **flowchart** to illustrate the steps required to complete the above task.
   ii. Write the **pseudo code** to accomplish the task described above
   iii. Write the **complete C program** that fulfills the above requirements

[30 marks]

c.

i. Write down **two (02)** differences between **primitive** and **derived** data types used in C programming.

[06 marks]

ii. Write down **two (02)** differences between **constants** and **variables** used in C programming.

[06 marks]

d. Write the **output** for each of the following code segments, assuming **x = 5** and **y = 10**. Provide **reasons** for obtaining each output.

i.
```
int z = y % x ;
printf("%d", z) ;
```

ii.
```
x *= y ;
printf("%d", x) ;
```

iii.
```
printf("%d", x != y) ;
```

iv. `printf("%d", ( !(x > 5) && (y < 15) ) ) ;`

v. `printf("%d", (++x) + (y++) ) ;`

vi.
```
int z = x + y * 2 - x % 2 / y ;
printf("%d", z) ;
```

vii.
```
float z = y / (x-1) ;
printf("%f", z) ;
```

[38 marks]

2.

a.

i. Briefly describe the relationship between **the ternary operator (? :)** and an **if–else** statement in C.

[04 marks]

ii. Write a C program to find the **largest of three integer numbers** using the **ternary operator.**
The program should accept three integers as user inputs and needs to display the largest number among them.
Sample input and output are shown below.

Input:	Output:
Enter three numbers: 12 45 23	Largest: 45

[15 marks]

b.

i. State whether the following statements are **TRUE** or **FALSE.**

- Each case label in a switch statement must have a unique constant value.
- The default case in a switch statement is optional.
- The expression inside a switch statement must evaluate to an integer or character type.
- You can use relational operators (like "<" or ">") in case labels.
- The continue statement can be used directly inside a switch statement outside of a loop.

[10 marks]

ii. Consider the following C program:

```
#include <stdio.h>

int main() {
   int choice;
   scanf("%d", &choice);

   switch (choice) {
      case 1:
         printf("One\n");
```

```
            break;

        case 2:
            printf("Two\n");

        default:
            printf("Default\n");
    }
    return 0;
}
```

Copy and complete the following table with the **outputs for each choice**, providing **reasons** for each output.

Input	Output	Reason/s
1		
2		
3		

[15 marks]

c.

i. What will be the **output** when you compile and run following C program. Provide **reasons** for obtaining the output.

```
#include <stdio.h>

int main() {
    for ( ; ; ) {
        printf("Hello\n");
    }
    return 0;
}
```

[05 marks]

ii. Consider the following C program:

```
#include <stdio.h>
int main() {
    int i = 1, j;

    for ( ; i <= 4; i++) {              // Outer loop
        for (j = 1; j <= i; j++) {      // Inner loop
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

1. Determine the **output** generated when the above C program is compiled and executed.
2. What will be the output if you replace **printf("* ");** with the following code segment?

```
if ( ( i % 2 ) == 1 )
    printf("* ");
```

3. What will be the **output** if you **swap the inner and outer loops** in the original code?

[15 marks]

d. Consider the following C program:

```
#include <stdio.h>

int main() {
   int i, out = 1;

   do {
      for ( i = 1 ; i <= out ; i++ ) {
         switch(i) {
            case 1:
               printf("X");
               continue;
            case 2:
               printf(" Y");
               continue;
            case 3:
               printf(" Z");
               break;
            default:
               break;
         }
      }

      out++;
      printf("\n");
   } while (out < 4);

   return 0;
}
```

Copy and complete the table below with the values of **out, i**, and **output** until the program terminates, providing **reasons** for each result.

out	i	output	reason/s

[36 marks]

3.

a.

i. Briefly describe the **difference** between a **user defined function** and a **recursive function** used in C programming.

[06 marks]

ii. Consider the following incomplete C program:

```
#include <stdio.h>
```

```
int main() {
    char name[50];

    printf("Enter your name: ");

    //Your code


    return 0;
}
```

Write the required code segment to allow the program to access the **user's full name as an input** and **display** it in the following format.

Hello <name>

Ex: user input → Nimal Perera
display → Hello Nimal Perera

[10 marks]

iii. Write a C program that **converts a decimal number** entered by the user into its **binary** form using a **recursive function**.

[14 marks]

b. Consider the following incomplete C program:

```
#include <stdio.h>

// Function to read marks into the 2D array
void readMarks

// Function to display total marks for each student
void displayTotals

int main() {
   int n; // Number of students

   printf("Enter number of students: ");
   scanf("%d", &n);

   // 2D array definition to store marks for 03 subjects of n students
   int marks

   // Call functions
   readMarks ( n, marks);
   displayTotals ( n, marks);

   return 0;
}
```

Complete the above C program to obtain the following output, according to the guidelines given below.

```
Enter number of students: 2

Enter marks for student 1 in 3 subjects: 80 70 90
Enter marks for student 2 in 3 subjects: 85 75 95

Student 1 Total: 240
Student 2 Total: 255
```

For that you need to:

- Create a **2D array** inside main() function to store marks for 03 subjects of user entered no of students.
- Write a user defined function named **readMarks** to read marks from the **console** and **store** marks into the **2D array**. You need to pass the number of students and marks into the function as parameters.
- Write a user defined function **displayTotals** to display the total marks for each student. You need to pass number of students and marks into this function.

[30 marks]

c.

i. Briefly describe the **difference** between **Address (&)** operator and **Contents (*)** operator used in C programming.

[06 marks]

ii. Briefly describe the **similarity** between a **pointer** and an **array name** in C.

[06 marks]

iii. Consider the following C program:

```
#include <stdio.h>

void swap(int x, int y) {
   int temp;
   temp = x;
   x = y;
   y = temp;
}

int main() {
   int a = 5, b = 10;

   printf("Before : a = %d, b = %d\n", a, b);

   swap(a, b);

   printf("After : a = %d, b = %d\n", a, b);

   return 0;
}
```

1. What **output** will be produced when the above C program is compiled and executed? Provide your **reason**(s).

2. What **code modifications** are required in the above program to produce the following output?

```
Before : a = 8, b = 4
After : a = 4, b = 8
```

[28 marks]

4.

a.

i. Briefly describe the **difference** between a **text file** and a **binary file** used in C programming.

[06 marks]

ii. Consider the following incomplete C program:

```
#include <stdio.h>
#include <string.h>

// Function to write data of 02 students to a file name student.txt
void writeData() {

  // Section A : Open file in write mode

  //Section B : Check for the file existence

  char name[100];     // Local variable for input name
  int age;            // Local variable for input age

  printf("\n Writing data to file:\n");

  //For loop to get two student details from the user
  for (int i = 0; i < 2; i++) {
    printf("Enter name of student %d: ", i + 1);

    getchar();        // Consume leftover newline from previous input

    // Section C : Read full name with spaces from user

    // Section D : Remove newline character at the end of name array

    printf("Enter age of student %d: ", i + 1);

    // Section E : Read age from user

    // Section F : Write name and age to file using comma as separator using name, age format

  }

  // Section G : Close the file
```

```
}

// Function to read data from file and display
void readData() {

  // Secation H : Open file in read mode

  //Section I : Check for the file existence


  char line[150];      // Buffer to read a line from file
  char name[100];      // Local variable for reading name
  int age;             // Local variable for reading age

  printf("\n Reading data from file:\n");

  // Section J : Read each line and parse name and age

  // Section K : Close the file

}
int main() {
  writeData(); // Call function to write student data
  readData(); // Call function to read and display data
  return 0;
}
```

Complete the C program to obtain the following output by completing **Section A** to **Section K.**

Writing data to file : Enter name of student 1: K Perera Enter age of student 1: 20 Enter name of student 2: N Fernando Enter age of student 2: 22	Reading data from file: Name : K Perera, Age : 20 Name : N Fernando, Age : 22

[44 marks]

b.

i. Briefly describe the **difference** between an **array** and a **struct** used in c programming.

[06 marks]

ii. Briefly describe the **difference** between a **union** and a **struct** used in c programming.

[06 marks]

iii. Consider the following incomplete C program:

```
#include <stdio.h>

// Section A : Define a struct to store 3 quiz marks as integer values

// Section B : Define a struct to store a student's registration number and marks

// Section C : Function to input data for 2 students
void storeStudents(                    ) {

  for (int i = 0; i < 2; i++) {  // Loop exactly 2 times for 2 students
    printf("Enter Registration Number for Student %d: ", i + 1); // Prompt for reg no

    // Section D : Read and store registration number

    printf("Enter marks for 3 quizzes: "); // Prompt for quiz marks

    // Section E : Read and store marks for the student

  }
}

// Section F : Function to display registration number and total marks for 2 students
void displayStudents(                    ) {

  for (int i = 0; i < 2; i++) {  // Loop exactly 2 times for 2 students

    // Section G : Calculate total marks for the student

    // Section H : Print reg no and total

  }
}

int main() {
  // Section I : Create an array to store 2 students

  storeStudents(students);    // Call function to input data for 2 students
  displayStudents(students);  // Call function to display data for 2 students

  return 0;  // Successful execution
}
```

Complete the C program to obtain the following output by completing **Section A** to **Section I**.

Sample input : Enter Registration Number for Student 1: TG1234 Enter marks for 3 quizzes: 15 18 20
Sample output : Reg No: TG1234, Total Marks: 53

[38 marks]

------ End of Paper ------