Tribhuvan University
Institute of Science and Technology

Course Title: Computer Programming
Course No: Math 404
Level: B. Sc
Nature of course: Theory (6 Hrs.) + Lab (3 Hrs.).

Full Marks: 100 (60 Theory + 40 Lab)
Pass Marks: 35 (21 Theory +14 Lab)
Year: IV
Period Per Week: 9 Hrs

Course Description: The course covers the basics of computer systems and programming in the "C" programming language. The course aims at demonstrating the fundamental programming techniques of C. The course includes basics of C-programming, scalar data types and their operators. The course spans the details of flow control, complex data types such as arrays, structures, unions, and pointers, structuring the code using functions, and handling the files.

Examination: There will be a final examination of the theory part of 60 marks for the period of two hours. The examination for the practical (laboratory) part of 40 marks will be conducted by the concerned Department of Mathematics and the marks will be submitted to office of the controller of examination. The candidate must pass in theory and practical (laboratory) part individually.

Detailed Course Contents:

Unit 1: Introduction to Computer Systems: 11 Hrs.

1.1 Introduction to computers: Definition
1.2 Architecture of digital computer: Central Processing Unit, Memory, Input/Output, Communication devices
1.3 The CPU: how it works (fetch, decode, execute and store cycle)
1.4 Memory system: Primary memory, Secondary memory,
1.5 Inputs / Output devices (For E.g. Mouse, Keyboard, Scanner, Display unit, Printer)
1.6 Computer software, Types of software: Application software, System software: Operating systems, Utility software,
1.7 Generations of computers: First, Second, Third, Fourth, and Fifth
1.8 Applications of computers.

Unit 2: Introduction to Programming Languages: 7 Hrs.

2.1 Program, Programming, Programming languages
2.2 Evolution of programming languages
2.3 Generations of programming languages: Machine-level, Assembly-level, and High-level
2.4 Structured programming, Object oriented programming
2.5 The compilation process, Source code, Object code, Executable code
2.6 Language processors: Interpreters, Compilers, Linkers, Loaders
2.7 Fundamentals of algorithms,
2.8 Flow charts.

Unit 3: Fundamentals of C Programming: 11 Hrs.

3.1 Introduction to C
3.2 History of C
3.3 Structure of C program, Compilation and execution of C Program, C-Library
3.4 The C-Character set, C-Tokens, Keywords and identifiers, Delimiters
3.5 Variables, Definition and declaration of variable, Constants, Escape Sequence Characters
3.6 Data types: Integer, Character, Float, Double,
3.7 Expressions, Statements, Comments, Symbolic constants.

Unit 4: Input/ Output Statements: **5 Hrs.**

4.1 Single character input/ output: *getchar* and *putchar* function
4.2 Input data using *scanf*, Conversion characters for data inputs
4.3 Writing output data using *printf*, Conversion characters for data output
4.4 *gets* and *puts* functions

Unit 5: Operators and Expressions: **10 Hrs.**

5.1 Arithmetic operators: Addition, subtraction, multiplication, division and remainder division
5.2 Unary operators: Unary minus (-), Increment operator, decrement operator
5.3 Relational operators (<, > , <= , >=), Equality operators (= =, !=), Logical operators (AND(&&), OR(||)), Assignment operator (=)
5.4 Conditional operator (? :), Bitwise operators (AND, OR, XOR, Shift Left, Shift Right, One's Complement), Comma operator
5.5 Precedence of operators
5.6 Arithmetic expressions
5.7 Type conversion in expressions

Unit 6: Control Statements: **8 Hrs.**

6.2 Introduction to control statements
6.3 Branching: Simple If statement, If else statement, Nested if-else statement, Else if ladder, Switch statement
6.4 Introduction to Looping statements
6.5 Looping using While statement, Do-while statement, and For statement
6.6 Break statement, Continue statement, Goto statement

Unit 7: Functions: **12 Hrs.**

7.1 Overview of functions,
7.2 Library functions,
7.3 User defined functions, Defining a function, Accessing a function, Function prototypes (Function Declaration),
7.4 Scope of the variables in a function: Local and Global variables,
7.5 Category of functions: functions with no arguments and no return values, functions with arguments and
no return values, functions with arguments and return values
7.5 Passing arguments to a function: Call-by-value, Call-by-reference
7.6 Recursion

Unit 8: Arrays and Pointers **18 Hrs.**

8.1 Introduction to array
8.2 Defining an array, Processing an array
8.3 Types of array: One-dimensional array, Two-dimensional array, Multi-dimensional array
8.4 Matrix operations using array
8.5 Arrays and strings
8.6 Introduction to pointers
8.7 The & and * operator, Declaring and initializing pointer, Accessing pointer
8.8 Passing pointers to a function
8.9 Pointers and arrays: (pointers and one dimensional array, pointes and multidimensional arrays)
8.10 Dynamic memory allocation
8.11 Operations on pointers
8.12 Array of pointers

Unit 9: Structures and Unions:
10 Hrs

9.1 Introduction to structures
9.2 Defining a structure, Processing a structure
9.3 User defined data type (Typedef)
9.4 Structures and Pointers
9.5 Passing structures to functions
9.6 Array of structures, Arrays within structure
9.6 Structure within structure (Nested/ self-referential Structure),
9.7 Unions

Unit 10: File Handling:
8 Hrs.

10.1 Concepts of file,
10.2 Opening a file using fopen and closing of file using fclose
10.3 Modes: read, write, append
10.4 Input/ output functions in file: getc, putc, getw, putw, fprintf, fscanf
10.5 Creating a file,
10.6 Processing a file

Laboratory works: **50 Hrs.**

This course requires a lot of programming practices. Each topic must be followed by a practical session. Practical sessions for each unit should be conducted and should include writing the C-programs. The instructors are highly encouraged to prepare lab sheets for individual units covering the mathematical problems as per the requirement. The sample lab sessions can be as following descriptions,

For Unit 1: **3 hours**

- Familiarize with computer system, hardware devices, software, operating system, working with files and folders

For Unit 3: **7 hours**

- Familiarize with c-programming language IDE.
- Write simple "Hello World" program, and familiarize with compiling process of C -program.
- Write the programs that will include the use of keywords, operators, escape sequence characters.
- Write programs that will illustrate the usages of the basic data types.

For Unit 4: **4 hours**

- Write the programs that include the usages of input-output functions like gets, puts, scanf, printf etc.
- Write the programs to illustrate the formatted input output sentences in C.

For Unit 5: **7 hours**

- Write programs to solve arithmetic expressions.

- Write programs like finding area of rectangle, area of circle, simple interest, solving some mathematical expressions etc.
- Write the programs that will reflect the usages of different types of operators mentioned in the unit 5.
- Write programs to illustrate type conversion.

For Unit 6: **6 hrs**

- Write programs using if else statements and switch statement to show control constructs for eg, writing programs for determining even or odd, checking positive and negative number, primality test etc.
- Write programs to illustrate looping constructs using while, do while and for. For examples finding sum on n natural numbers, determining Armstrong number, finding factorial of a number, generating Fibonacci series, determining whether a number is palindrome or not, generating prime numbers from 1 to 100, generating binomial coefficients etc.
- Write programs to show the use of break, continue and goto statements..

For Unit 7: **7 hrs**

- Write programs using C-library functions.
- Write programs to solve different mathematical problems using functions to realize the modular programming. The above mentioned programs can be implemented using functions. For eg. program for finding GCD using Euclid's Algorithm, finding factorial, finding Fibonacci series of any given number, quadratic roots of equation etc.
- Write programs using functions to realize call by reference and call by value.
- Write programs with functions having certain return types.
- Write programs using user defined recursive functions.

For Unit 8: **8 hrs**

- Write programs for realizing user defined arrays (one-dimensional, multidimensional)
- Write programs using arrays for creating matrices, performing operations on matrices like addition, subtraction, multiplication of matrices.
- Write program to implement adjacency graph using arrays.
- Write programs to realize use of pointers. (The above mentioned programs can be implemented by using pointers)

For Unit 9: **5 hrs**

- Write programs for creating user defined data types using Structures and Unions.

For Unit 10: **3 Hrs**

- Write programs for creating files.
- Write programs to read, write content in a file.

Text/ References Books:

1. Byron S. Gottefried, " *Theory and Problems of Programming with C*", Mc-Graw Hill.
2. Brian W. Kernighan & Dennis M. Ritchie, "*The C programming Language*", PHI.
3. E. Balagurusamy, "*Programming in ANSI C*", Tata Mc-Graw Hill.
4. Yashavant Kanetkar: "*Let us C*", BPB Publications.
5. Stephen G. Kochan, "*Programming in C*", CBS Publishers & Distributors.
6. Efraim Turban, R. Kelly Rainer, Jr. Richard E. Potter, "*Introduction to Information Technology",* John Wiley & Sons (Asia) Pvt. Ltd.
7. Alexis Leon, Mathews Leon, "*Fundamentals of Information Technology*", Leon TechWorld.

Guidelines to the question setters

The question paper setters are requested to:

- prepare the questions such that it should cover almost every units of the course. The questions should be fairly distributed over the whole course as per the working hours mentioned.

- review that the question selected should not be too short to be answered within a few minutes or not be too long. The time allocated for each question is 20 minutes.

- see that the paper set shall be such as candidates can reasonably be expected to answer within the time allotted. So the questions should consist of both theoretical components as well as writing c-programs.

- make sure that the questions are within the capacity of candidates.

- review that the questions are framed in such a way that there shall not be any ambiguity and repetition.

- make sure that the choice question can be attached to any question and can be framed from any unit

On the basis of the guidelines mentioned, we enclose one set of model question for Computer Programming (Math 404)

Model question

Tribhuvan University

Bachelor Level / IV year/ Sc. & Tech. Full Marks: 60

Computer Programming (Math 404) Time: 2 Hours

Attempt all Questions. All Questions carry equal marks.

1. What is the use of CPU in a computer system? How it works? How variables in a c-program can be defined? Support your answer by writing a c-program containing declaration and definition of variables. [2+3+5]
   [Unit 1 and Unit 3]

2. Why language processor is needed? Differentiate compilers from interpreters. Write a program to capture inputs and display outputs for variables name, address and roll_number. Use different escape sequence characters for displaying the formatted output. [2+3+5]
   [Unit 2 and Unit 4]

3. Define the different types of logical operators used in C. Write a program using function for calculating total distance travelled by a vehicle in t seconds, where the distance is given by the expression
   $$\text{distance} = ut + (at^2) / 2$$
   The program should provide flexibility to capture inputs for the variables u (initial velocity), a (acceleration) and the time (t). [4+6]
   [Unit 5 and Unit 7]

4. What is the difference between the while and do while statements? How continue statement can be used in while loop? How fopen and fclose statements can be used in file handling? [4+2+4]
   [Unit 10 and unit 6]

5. What is a pointer? How pointer is initialized? Write a program to read two arrays, add the two arrays and store the result in third array. Print the final output of the third array. Label your program with proper comments. [2+2+6]
   [Unit 8]

6. Discuss about the possible approaches for passing arguments to a function. How structure differs from union? How can you create array of structures? [4+4+2]

   OR

   Define function category with parameters and return value. How recursion works? Illustrate with suitable example. How self-referential structures can be defined? [2.5+2.5+5]
   [Unit 7 and 9]