Course Booklet: C Programming

1. Course Details

- **Course Code**: CSC115
- **Title**: C Programming
- **Credits**: 3
- **Program Name**: Bachelor of Science in Computer Science and Information Technology
- **School**: Institute of Science and Technology
- **Year**: 1
- **Semester**: 1
- **Course Leader**: Er. Sharat Maharjan
- **Email**: sharat.maharjan@prime.edu.np
- **Office Hours**: 6:30am to 11:30am [Sunday – Friday]
- **Session**: AY2081/82
- **Session Start Date**: Mangsir 2081
- **Planned Hours**: 60 [in class – 48 hours, 6 hours of assessment support, 6 hours of self-study]
- **Virtual Learning Environment (VLE)**: MS Teams
- **Timetable**: Refer to Published Timetable

2. Learning Outcomes of the Course

After completing this course, students will be able to:

1. Analyze and solve computational problems using structured programming techniques.
2. Demonstrate proficiency in C programming concepts such as variables, control structures, functions, pointers, and file handling.
3. Develop efficient algorithms and debug C programs.

3. Skills and Knowledge Gained after Completing this Course

Skills	Knowledge
Problem-Solving	Programming Fundamentals
Debugging	Control Structures
Algorithm Design	Data Manipulation
C Programming	File Operations

4. Detailed Course Content

- **Unit 1: Problem Solving with Computer** - 2 Hrs
  Problem analysis, Algorithms and Flowchart, Coding, Compilation and Execution, History of C, Structure of a C program, Debugging, Testing and Documentation.

- **Unit 2: Elements of C** - 4 Hrs
  C Standards (ANSI C and C99), Character Set, Tokens, Escape sequences, Data types, Constants, Expressions, Statements, Structure of a C program.

- **Unit 3: Input and Output** - 2 Hrs
  Conversion specification, Reading and Writing characters, I/O operations, Formatted I/O.
- **Unit 4: Operators and Expression** - 4 Hrs
  Various operators (Arithmetic, Logical, Bitwise), Expressions, Operator Precedence and Associativity.
- **Unit 5: Control Statement** - 4 Hrs
  Conditional Statements, Loops, Break, Continue, Exit function.
- **Unit 6: Arrays** - 6 Hrs
  Array types, Memory Representation, Strings, String Library Functions (e.g., length, copy, concatenation).
- **Unit 7: Functions** - 5 Hrs
  Library and User-defined functions, Function Arguments, Recursion, Scope visibility.
- **Unit 8: Structure and Union** - 5 Hrs
  Array of structures, Nested Structure, Union, Pointers to structures.
- **Unit 9: Pointers** - 6 Hrs
  Pointer Arithmetic, Dynamic Memory Allocation, Function Pointers, Pointers with Arrays and Strings.
- **Unit 10: File Handling in C** - 4 Hrs
  File operations, Random access, Error Handling in Files.
- **Unit 11: Introduction to Graphics** - 3 Hrs
  Concepts of Graphics, Graphics Initialization, Basic Graphic Functions.

---

5. Course Lesson Delivery

Unit	Week	Topics Covered	Session Type
Unit 1: Problem Solving with Computer	Week 1	Introduction to problem analysis, algorithms, flowcharts, C program structure	Lecture, Practical
Unit 2: Elements of C	Week 2	C standards, tokens, variables, data types, program structure	Lecture, Workshop
Unit 3: Input and Output	Week 3	I/O operations, character reading/writing, formatted I/O	Lecture, Tutorial
Unit 4: Operators and Expression	Week 4	Operator types, expressions, precedence, associativity	Lecture, Lab Work
Unit 5: Control Statement	Weeks 4-5	Conditional statements, loops, exit function, break/continue	Lecture, Group Activity
Unit 6: Arrays	Weeks 6-7	Array concepts, memory, strings, string library functions	Lecture, Hands-on Practice

Unit 7: Functions	Week 8	Library and user-defined functions, recursion, argument passing	Lecture, Practical Work
Unit 8: Structure and Union	Week 9	Structures, unions, arrays of structures, pointer to structures	Lecture, Case Study
Unit 9: Pointers	Week 10	Pointer basics, arithmetic, dynamic memory, pointers with arrays	Lecture, Lab
Unit 10: File Handling in C	Week 11	File operations, random access, error handling	Lecture, Practical Work
Unit 11: Introduction to Graphics	Week 12	Graphics functions, initialization, drawing shapes	Lecture, Workshop

---

6. Important Dates (Exam/Submission/Deadlines)

Week	Type	Event
8	Midterm Exam	Theory Exam - 2 Hours
13	Final Project	Programming Project Due
15	Final Exam	Comprehensive Practical and Theory Exam - 3 Hours

---

7. Teaching and Learning Methods

This course is delivered through a blend of lectures, tutorials, and practical lab sessions. Key approaches include:

- **Lectures**: Structured sessions covering core programming concepts.
- **Tutorials**: Focused sessions for addressing specific questions and clarifying complex topics.
- **Practical Labs**: Hands-on programming exercises to apply theoretical knowledge.

---

8. Internal Assessment and Evaluation

Type	Title	Weight
Assignment	Programming Exercises and Lab Work	40%
Class Participation	In-class Coding Challenges	10%
Midterm Exam	Written Theory Exam	25%
Final Exam	Comprehensive Practical Exam	25%

---

9. Academic Policies

- **Attendance**: A minimum of 85% attendance is required to maintain eligibility for final assessments.
- **Late Submission**: Late work may incur penalties unless prior arrangements are made.

- **Academic Integrity**: Plagiarism and cheating are strictly prohibited. Maintain honesty in all submissions.

---

10. Resources and Support

- **Required Textbooks**:
  - *Programming with C* by Byron Gottfried
  - *The C Programming Language* by Kernighan and Ritchie
- **Online Resources**: Access additional materials and lecture notes on MS Teams.
- **Student Support Services**: Contact your faculty advisor or academic counselor for guidance.

---

11. Student Responsibilities

Students are expected to:

- **Participate Actively**: Engage in classes, labs, and group activities.
- **Manage Time Effectively**: Prioritize academic commitments and meet deadlines.
- **Uphold the Code of Conduct**: Follow classroom rules and respect peers and faculty.

---

12. Approved Coursework for this Session

Assessment	Title	Week Released	Submission Date	Learning Outcomes
Assignment	Programming Project	Week 2	Week 13	Problem-solving, Programming Skills

---

13. Appendices

Appendix A: TU Official C Programming Syllabus

Tribhuvan University

Institute of Science and Technology

2080

Bachelor Level / First Year / First Semester — Full Marks: 60

Computer Science and Information Technology (CSC115) — Pass Marks: 24

(C Programming) Model Question — Time: 3 hours

Candidates are required to give answers in their own words and as far as practicable.

Section A

Attempt any two questions. (2 × 10 = 20)

1. Define function and list its advantages. Describe the difference between passing arguments by value and passing arguments by address with suitable program. [4+6]
2. Explain how structure is different from union? Make a program using structure of booklist having data member's title, author, and cost. Enter four data and calculate total cost. [3+4+3]
3. Explain various modes in which file can be opened? Write a program to CREATE and WRITE N numbers in a file "NUMBER.TXT". Open this file then read its content and put all even numbers in one file "EVEN.TXT" and odd numbers in another file "ODD.TXT". [2+4+4]

Section B

Attempt any eight questions. (8 × 5 = 40)

4. What do you mean by a problem analysis? What are the properties of a good algorithm? Explain the Compilation and Execution of any C program? [1+1+3]
5. Define nested if else statement with suitable flowchart. Write a C code to check if user given input is exactly divisible by 5 or 11 using nested if else statement? [2+3]
6. List various binary and unary operators used in C? Write a program that uses a "while" loop to compute and prints the sum of a given numbers of squares. For example, if 4 is input, then the program will print 30, which is equal to $1^2+2^2+3^2+4^2$. [1+4]
7. "Size of character array is always declared one more than the input size." Justify the statement. Write a program to read a character array input as "TRIBHUVAN UNIVERSITY" from the user and find out how many times a character 'I' occurs in that array? [1+4]
8. Write syntax to declare and initialize 2-dimensional array? With suitable program logic explain how would you find transpose of a 3*3 matrix? [1+4]

9. Explain the concept of recursive function using the example program to find the factorial of given positive integer. [5]

10. Describe the fundamental concept of pointer and its arithmetic with suitable examples. [5]

11. Explain the use of graphical functions. Write a program to draw a triangle using line() graphics function. [1+4]

12. Write short notes on: [2+3]

    i) Dynamic Memory Allocation

    ii) break and continue