

CPS-106: COMPUTER FUNDAMENTALS & C PROGRAMMING

Course Title:	Computer Fundamentals & C Programming	Course Code	CPS-106
Year	1st YEAR	Course Group	CORE
Teaching Scheme in Periods (L:T:P)	4:1:0	Credits	3
Methodology	Lecture + Tutorials	Total Contact Periods	150
CIE	60 Marks	SEE	40 Marks

Pre requisites

Basic knowledge Of Mathematics, Logical, Reasoning and IT skills

Course Outcomes

Upon completion of the course, the student shall be able to

CO1	Apply algorithmic thinking to solve computational problems and explain the fundamental building blocks of embedded systems
CO2	Select appropriate data types and operators to construct simple C expressions for mathematical calculations.
CO3	Develop logic using control structures (decision-making and loops) to solve iterative and conditional problems.
CO4	Construct modular and efficient programs using functions and suitable storage classes.
CO5	Implement derived data types such as arrays, strings, and structures to organize and manipulate data effectively.
CO6	Utilize pointers for direct memory access and bitwise operators for low-level data manipulation required in embedded applications.

Course Content and Blue Print of Marks for SEE

Unit No	Unit Name	Periods	Questions to be set for SEE			
			R	U	A	
I	Computer Fundamentals and Problem Solving	20	Q4	Q1	Q9(a)	Q13(a)
II	Fundamentals of C Language	20				
III	Control Structures	28		Q2	Q10(a)	Q14(a)
IV	Functions and storage classes	28				

V	Arrays, Strings, and User-Defined Data Types	26	Q3	Q5,Q6	Q9(b),Q11(a), Q11(b)	Q13(b),Q15(a), Q15(b)
VI	Pointers, Memory, and Bitwise Operations	28		Q7,Q8	Q10(b),Q12(a), Q12(b)	Q14(b),Q16(a), Q16(b)
	Total	150	8		8	8

Course Contents

UNIT-1: Computer Fundamentals and Problem Solving

Duration: 20 Periods (L: 16 – T:4)

Introduction to Computers – characteristics and history (generations) – block diagram of a digital computer – functions of CPU, ALU, Control Unit, and I/O devices – memory organization – memory hierarchy (Registers, Cache, RAM, ROM) – measurement units of memory – classification of computers – supercomputers, mainframes, workstations, PCs Problem solving using computers – steps in problem solving – algorithm – characteristics of algorithm – writing algorithms for simple problems – flowchart – flowchart symbols – rules for drawing flowcharts – flowcharts for decision making and looping – program development cycle – problem definition – analysis – design – coding – compilation – execution – testing and debugging – introduction to programming languages – machine language – assembly language – high-level languages – compiler and interpreter – introduction to embedded systems – definition of embedded system – general block diagram of embedded system – examples of embedded systems – role of software in embedded systems – role of C language in embedded applications.

UNIT-2: Fundamentals of C Language

Duration: 20 Periods (L: 16 – T:4)

Structure of C program – C character set – tokens – keywords – identifiers – data types – size and range of data types – Introduction to fixed-width integers (<stdint.h>) – variables – declaration and initialization of variables – constants – operators – arithmetic operators – relational operators – logical operators – assignment operators – sizeof operator – increment and decrement operators – expressions – operator precedence and associativity – input and output functions – printf() – scanf() – format specifiers – type conversion – implicit type conversion – explicit type conversion.

UNIT-3: Control Structures

Duration: 28 Periods (L: 23 – T:5)

Control statements – decision making statements – if statement – if-else statement – nested if – else-if ladder – switch-case statement – looping statements – for loop – while loop – do-while loop – Infinite loops (while(1)) – jump statements – break statement – continue statement – goto statement – menu-driven programs – pattern generation programs – logical and iterative applications.

UNIT-4: Functions and storage classes

Duration: 28 Periods (L: 23 – T:5)

Need for functions – advantages of functions – modular programming – types of functions – library functions – user-defined functions – function declaration – function definition – function call – passing arguments to functions – call by value – call by reference

(introduction) – return statement – recursion – difference between recursion and iteration – storage classes – auto – static – extern – register – header files – preprocessor directives – macros – #define.

UNIT-5: Arrays, Strings, and User-Defined Data Types

Duration: 26 Periods (L: 21 – T:5)

Arrays – one-dimensional arrays – two-dimensional arrays – array initialization – array applications – strings – string declaration and initialization – string input and output – string handling functions – user-defined data types – structures – structure declaration and initialization – accessing structure members – array of structures – unions – difference between structure and union – enumerations – enum keyword – applications of enumerations.

UNIT-6: Pointers, Memory, and Bitwise Operations

Duration: 28 Periods (L: 23 – T:5)

Memory concepts – memory organization – RAM and ROM – pointers – pointer declaration and initialization – pointer arithmetic – pointers and arrays – pointers and functions – dynamic memory allocation – malloc() – calloc() – realloc() – free() – bitwise operators – bitwise AND – OR – XOR – NOT – left shift – right shift – bit manipulation techniques – setting a bit – clearing a bit – toggling a bit – checking a bit.

Text Books

- | | | |
|-----------------------------|----|-------------------------------------|
| 1. Let Us C | -- | Yeshwanth Kanetkar BPB Publications |
| 2. Programming in ANSI C | -- | E. Balaguruswamy Tata McGrawHill |
| 3. Programming with C | -- | Gottfried Schaum's outline |
| 4. C The complete Reference | -- | Schildt Tata McGraw Hill |

Reference Books

1. The C Programming Language- Brian W. Kernighan, Dennis M. Ritchie- Pre Publicationntice Hall
2. Programming in C 3rd Edition-Sams Publishing- Steven G. Kochan- Sams Publishing
3. Expert C Programming: Deep Secrets- Peter Van Der Linden- Pearson Publications
4. C Programming: A Modern Approach. K.N. King- Norton Publications
5. Learn to Program with C- Noel Kalicharan- APress Publications

Suggested e-Learning Links

1. https://www.w3schools.com/c/c_intro.php?external_link=true
2. <https://www.geeksforgeeks.org/c-programming-language/>
3. <https://www.programiz.com/c-programming>
4. <https://www.javatpoint.com/c-programming-language-tutorial>
5. <https://www.freecodecamp.org/news/the-c-programming-handbook-for-beginners/>
6. <https://www.tutorialspoint.com/cprogramming/index.htm>
7. <https://www.codecademy.com/catalog/language/c>
8. <https://www.codechef.com/learn/course/c>
9. <https://archive.nptel.ac.in/courses/106/104/106104128/>
10. <https://www.youtube.com/watch?v=3QiltmIWmOM>

11. <https://www.digimat.in/nptel/courses/video/106105171/L01.html>
12. <https://www.digimat.in/nptel/courses/video/106105171/L07.html>

Suggested Learning Outcomes

For achieving the Course outcomes, the following learning outcomes must be achieved

Suggested Learning Outcomes:

Upon completion of the Course, the student shall be able to

CO1: Apply algorithmic thinking to solve computational problems and explain the fundamental building blocks of embedded systems.

- 1.1 Define a digital computer and list its key characteristics like speed, accuracy, and versatility.
- 1.2 Draw the block diagram of a digital computer showing the CPU, Memory, and I/O units.
- 1.3 Explain the specific functions of the ALU, Control Unit, and CPU Registers.
- 1.4 Summarize the evolution of computers through the five generations of hardware technology.
- 1.5 Classify computers into types like Supercomputers, Mainframes, Workstations, and PCs.
- 1.6 Differentiate between Volatile Memory (RAM) and Non-Volatile Memory (ROM).
- 1.7 Illustrate the memory hierarchy from Registers to Cache, RAM, and Secondary Storage.
- 1.8 Convert data storage units between Bits, Bytes, KB, MB, GB, and TB.
- 1.9 List the standard Input and Output devices used in modern computing systems.
- 1.10 Explain the sequential steps involved in Computer Problem Solving (Analysis to Testing).
- 1.11 Define an Algorithm and list its essential characteristics (Finiteness, Definiteness, etc.).
- 1.12 Write algorithms for simple arithmetic problems like calculating Interest or Swapping numbers.
- 1.13 Identify the standard ANSI symbols used for drawing Flowcharts.
- 1.14 Draw flowcharts to represent decision-making logic (e.g., Even/Odd check).
- 1.15 Draw flowcharts to represent looping logic (e.g., Sum of N numbers).
- 1.16 Explain the stages of the Program Development Cycle from source code editing to execution.
- 1.17 Compare Machine Language, Assembly Language, and High-Level Languages.
- 1.18 Distinguish between the working mechanisms of a Compiler and an Interpreter.
- 1.19 Define an Embedded System and sketch its general block diagram.
- 1.20 Justify the preference for C language in developing embedded firmware.

CO2: Select appropriate data types and operators to construct simple C expressions for mathematical calculations.

- 2.1 Explain the basic structure of a C program.
- 2.2 Identify the components of a C program.
- 2.3 Identify C character set, tokens, keywords, and identifiers.
- 2.4 Classify data types used in C language.
- 2.5 Determine the size and range of different data types.
- 2.6 Explain the concept of fixed-width integer types using <stdint.h>.
- 2.7 Select appropriate fixed-width data types (e.g., uint8_t) for embedded-oriented programs.

- 2.8 Declare and initialize variables correctly.
- 2.9 Identify and use constants in programs.
- 2.10 Apply arithmetic operators in expressions.
- 2.11 Apply relational and logical operators in decision making.
- 2.12 Use assignment, increment, and decrement operators.
- 2.13 Use the sizeof operator to determine memory usage.
- 2.14 Evaluate expressions using operator precedence and associativity.
- 2.15 Perform input operations using scanf().
- 2.16 Perform output operations using printf().
- 2.17 Use correct format specifiers for different data types.
- 2.18 Explain implicit type conversion.
- 2.19 Apply explicit type conversion using type casting.
- 2.20 Write simple C programs using fundamental language constructs.

CO3: Develop logic using control structures (decision-making and loops) to solve iterative and conditional problems.

- 3.1 Explain the need for control structures in programming.
- 3.2 Apply if statement for simple decision making.
- 3.3 Apply if–else statement for alternative execution paths.
- 3.4 Use nested if statements for multiple conditions.
- 3.5 Apply else-if ladder for multi-condition problems.
- 3.6 Use switch–case statement for menu-driven programs.
- 3.7 Differentiate between for, while, and do-while loops.
- 3.8 Use for loop for fixed-iteration problems.
- 3.9 Use while loop for conditional repetition.
- 3.10 Use do-while loop for at-least-once execution.
- 3.11 Implement infinite loops using while(1).
- 3.12 Justify the necessity of the "Super Loop" architecture in embedded systems.
- 3.13 Apply break statement to terminate loops.
- 3.14 Apply continue statement to skip loop iterations.
- 3.15 Explain the purpose of the goto statement.
- 3.16 Identify limitations of using goto.
- 3.17 Develop menu-driven programs using control structures.
- 3.18 Develop pattern generation programs.
- 3.19 Develop logical and iterative programs.
- 3.20 Debug logical errors in control structure programs.

CO4: Construct modular and efficient programs using functions and suitable storage classes.

- 4.1 Explain the need for functions in program development.
- 4.2 List the advantages of modular programming.
- 4.3 Differentiate between library and user-defined functions.
- 4.4 Declare user-defined functions.

- 4.5 Define and call functions in C programs.
- 4.6 Pass arguments to functions using call by value.
- 4.7 Explain the concept of call by reference.
- 4.8 Use return statement to pass values from functions.
- 4.9 Develop programs using multiple functions.
- 4.10 Explain recursion with suitable examples.
- 4.11 Differentiate between recursion and iteration.
- 4.12 Identify different storage classes in C.
- 4.13 Explain the scope and lifetime of variables.

- 4.14 Demonstrate the data persistence capability of the static storage class.
- 4.15 Explain the use of extern and register.
- 4.16 Use header files in C programs.
- 4.17 Explain the role of preprocessor directives.
- 4.18 Define macros using #define.
- 4.19 Apply macros in simple programs.
- 4.20 Organize C programs using modular techniques.

CO5: Implement derived data types such as arrays, strings, and structures to organize and manipulate data effectively.

- 5.1 Explain the need for arrays in programming.
- 5.2 Declare and initialize one-dimensional arrays.
- 5.3 Access elements of one-dimensional arrays.
- 5.4 Declare and initialize two-dimensional arrays.
- 5.5 Apply arrays in solving data storage problems.
- 5.6 Explain the concept of strings in C.
- 5.7 Declare and initialize strings.
- 5.8 Perform string input and output operations.
- 5.9 Apply standard string handling functions.
- 5.10 Differentiate between strings and character arrays.
- 5.11 Define structures and bit-fields to organize data efficiently.
- 5.12 Declare and initialize structures.
- 5.13 Access structure members using dot operator.
- 5.14 Use arrays of structures.
- 5.15 Explain the concept of unions.
- 5.16 Differentiate between structures and unions.
- 5.17 Define enumerations using enum.
- 5.18 Apply enumerations in programs.
- 5.19 Organize related data using user-defined data types.
- 5.20 Develop simple applications using arrays, strings, and structures.

CO6: Utilize pointers for direct memory access and bitwise operators for low-level data manipulation required in embedded applications.

- 6.1 Explain basic memory concepts.

- 6.2 Describe memory organization in a computer system.
- 6.3 Differentiate between RAM and ROM.
- 6.4 Define pointers and explain their purpose.
- 6.5 Declare and initialize pointers.
- 6.6 Access data and memory addresses directly using pointers.
- 6.7 Apply pointer arithmetic.
- 6.8 Explain the relationship between pointers and arrays.
- 6.9 Use pointers with functions.
- 6.10 Explain the need for dynamic memory allocation.
- 6.11 Identify the usage of malloc() and calloc().
- 6.12 Identify the usage of realloc() and free().
- 6.13 Explain memory management concepts.
- 6.14 Define bitwise operators.
- 6.15 Apply bitwise AND, OR, XOR, and NOT operations.
- 6.16 Apply left shift and right shift operations.
- 6.17 Perform bit setting operations.
- 6.18 Perform bit clearing operations.
- 6.19 Perform bit toggling and bit checking operations.
- 6.20 Relate bitwise operations to embedded system programming concepts.

Suggested Student Activities

Student activity like mini-project, surveys, quizzes, etc. should be done in group of 5-10 students.

1. Each group should do any one of the following type of activity or any other similar activity related to the course with prior approval from the course coordinator and programme coordinator concerned.
2. Each group should conduct different activity and no repetition should occur.
3. Explore and analyse topics to improve the level of creativity and analytical skill by taking Quiz/ tests/ assignments. Documents have to be maintained as a record.
4. Create a power point presentation on the topic relevant to course or advanced topic as an extension to the course to improve the communication skills. Documents have to be maintained as a record.
5. Visit different sites relevant to topics. Listen to the lectures and submit a handwritten report
6. Coding competitions

CO-PO Mapping Matrix

	Basic and Discipline Specific Knowledge	Problem Analysis	Design/Development of Solutions	Engineering Tools, Experimentation and Testing	Engineering Practices for Society, Sustainability and Environment	Project Management	Lifelong Learning	Linked PO
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	
CO1	3	2	1	1			2	1,2,3,4,7
CO2	3	2					1	1,2,7
CO3	3	2	1	1			2	1,2,3,4,7
CO4	3	2					1	1,2,7
CO5	3	2					1	1,2,7
CO6	3	3	2	2			3	1,2,3,4,7

QUESTION PAPER PATTERN FOR MID SEMESTER EXAMS

S.No	Description	Level	No of Questions	Marks for each question	Choice	Total Marks
01	Part-A	Remembering(R)	4	1	Nil	4 Marks
02	Part-B	Understanding(U)	4	3	2	6 Marks
03	Part-C	Application(A)	4	5	2	10 Marks
Total Marks						20 Marks

MID SEM-I EXAM

S.No	Unit No	R	U	A	Remarks
1	Unit-I	1,2	5(a)	7(a)	
			5(b)	7(b)	
2	Unit-II	3,4	6(a)	8(a)	
			6(b)	8(b)	
Total Questions		4	4	4	

MID SEM-II EXAM

S.No	Unit No	R	U	A	Remarks
1	Unit-III	1,2	5(a)	7(a)	
			5(b)	7(b)	
2	Unit-IV	3,4	6(a)	8(a)	
			6(b)	8(b)	
Total Questions		4	4	4	

The length of answer for each question framed in respect of Part-A, B&C shall not exceed ¼ of a page, 1 page and 2 pages respectively

QUESTION PAPER PATTERN FOR SEMESTER END EXAM

S.No	Description	Level	No of Questions	Marks for each question	Choice	Total Marks
01	Part-A	Remembering(R)	8	1	Nil	8 Marks
02	Part-B	Understanding(U)	8	3	4	12 Marks
03	Part-C	Application(A)	8	5	4	20 Marks
Total Marks						40 Marks

Unit No	Questions to be set for SEE				
	R			U	A
I	Q4	Q1		Q9(a)	Q13(a)
II					
III		Q2		Q10(a)	Q14(a)
IV					
V		Q3	Q5,Q6	Q9(b), Q11(a), Q11(b)	Q13(b), Q15(a), Q15(b)
VI			Q7,Q8	Q10(b), Q12(a), Q12(b)	Q14(b), Q16(a), Q16(b)
Total Questions	8			8	8

State Board of Technical Education and Training, Telangana
Model Question paper
Mid Semester-I Examination

Course Code: CPS-106

Course Name: Computer Fundamentals & C Programming

Duration: 1 hour

Max. Marks: 20 Marks

PART-A

Answer **all** questions- Each Question carries **ONE** mark
Marks

4x1 = 4

1. Define a "Digital Computer".
2. What is the full form of ALU?
3. What is the size of uint8_t in bits?
4. Which format specifier is used for printing a float value?

PART-B

Answer **two** questions- Each Question carries **THREE** marks
Marks

2x3 = 6

5.a) Draw the block diagram of a Digital Computer.

(OR)

b) List the differences between RAM and ROM

6.a) Explain the structure of a C program with an example.

(OR)

b) Differentiate between Implicit and Explicit Type Conversion.

PART-C

Answer **two** questions- Each Question carries **FIVE** marks

2x5 = 10 Marks

7a) Explain the different Generations of Computers with their key technologies and examples.

(OR)

b) Draw and explain the general block diagram of an Embedded System.

8a) Explain the different data types available in C along with their memory size and range.

(OR)

b) Discuss the various types of Operators in C with suitable examples.

State Board of Technical Education and Training, Telangana
Model Question paper
Mid Semester-II Examination

Course Code: CPS-106

Duration: 1 hour

Course Name: Computer Fundamentals & C Programming
Marks

Max.Marks: 20

PART-A

Answer **all** questions- Each Question carries **ONE** mark
Marks

4x1 = 4

1. What is the syntax of an infinite while loop?
2. Which keyword is used to skip the current iteration of a loop?
3. What is a Function Prototype?
4. Which keyword is used to return a value from a function?

PART-B

Answer **two** questions- Each Question carries **THREE** marks
Marks

2x3 = 6

- 5a) Differentiate between while and do-while loops.
(OR)
- 5b) Explain the syntax and working of the switch-case statement.
- 6a) List the advantages of using functions in C.
(OR)
- 6b) Differentiate between Call by Value and Call by Reference.

PART-C

Answer **two** questions- Each Question carries **FIVE** marks

2x5 = 10 Marks

- 7a) Write a C program to find the Sum of Digits of a given number using a while loop.
(OR)

7b) Explain the else-if ladder with a program to grade a student based on marks.

8a) Explain the four different Storage Classes (auto, extern, static, register) with examples.
(OR)

8b) Write a C program to find the Factorial of a number using Recursion.

State Board of Technical Education and Training, Telangana
Model Question paper
Semester End Examination

Course Code: CPS-106

Course Name: Computer Fundamentals & C Programming
Marks

Duration: 2 hour

Max.Marks: 40

PART-A

Answer **all** questions- Each Question carries **ONE** mark

8x1 = 8 Marks

1. Name one example of a Non-Volatile memory.
2. Define "Recursion".
3. Which function is used to free dynamically allocated memory?
4. Define "Enumeration" (enum).
5. What is the size of a Structure variable?
6. Which operator is used to access structure members?
7. What is the result of $1 \ll 2$?
8. Write the bitwise XOR operator symbol.

PART-B

Answer **Four** questions- Each Question carries **THREE** marks

4x3 = 12 Marks

9a) Write an algorithm to swap two numbers.

(OR)

9b) Explain how to initialize a 2D array with an example.

10a) Write a C program to check if a number is Even or Odd using if-else.

(OR)

10b) Write a C snippet to set the 3rd bit of a variable x to 1.

11a) List any three String handling functions from `<string.h>` and their uses.

(OR)

11b) Define a Structure for a Student with roll number and name.

12a) Explain Pointer Arithmetic with a simple example.

(OR)

12b) Differentiate between Stack and Heap memory.

PART-C

Answer **Four** questions- Each Question carries **FIVE** marks

4x5 = 20 Marks

13a) Draw a Flowchart to find the largest of three numbers. Explain the symbols used.

(OR)

13b) Write a C program to check if a given string is a Palindrome

14a) Write a C program to check whether a given number is Prime or not.

(OR)

14b) Explain Dynamic Memory Allocation functions (malloc, calloc, realloc, free) with syntax.

15a) Explain the concept of "Array of Structures" with a program to store details of 3 students

(OR)

15b) Discuss standard String library functions (strlen, strcpy, strcat, strcmp) with examples.

16a) Discuss Bit Manipulation techniques: Setting, Clearing, and Toggling a specific bit in a register.

(OR)

16b) Explain the relationship between Pointers and Arrays with a program to access array elements using pointers.