SCHOOL OF ENGINEERING

C PROGRAMMING FOR PROBLEM SOLVING

COURSE OVERVIEW

By, **Dr. SRIDHAR S K/Dr. N Bharathiraj**, Course Coordinator(s)

GENERAL INSTRUCTIONS

SCHOOL OF ENGINEERING

- **LAPTOPS ONLY**, NO MOBILE PHONES for Students.
- Laptops should only be used for **class-related tasks.**
- Students must **follow policy to ensure compliance**.
- Focus on **Course plan, Course outcomes, CO-PO matrix.**
- Follow Prescribed **Text books** for examination.
- Be aware of **CIA marks distribution** for the course.

SPECIFIC INSTRUCTIONS

SCHOOL OF ENGINEERING

- Learn Problem solving skills
- Learn Good Coding Practices.
- Learn algorithms and flowcharts before actual coding.
- Understand the practical applications of programming.
- Participate actively in coding challenges and quizzes

COURSE SYLLABUS

SCHOOL OF ENGINEERING

UNIT – I	07 Hours
Basics and overview of C: Introduction to Problem Solving using Algorithms and Flowchart: Key features of Algorithms: Sequence, Decision, Repetition with examples. Background, structure of C program, keywords, Identifiers, Data Types, Variables, Constants, Input / Output statements, Operators (Arithmetic, relational, logical, bitwise etc.), Expressions, Precedence and Associativity, Expression Evaluation, Type conversions. Conditional Branching Statements-if and switch statements, iterative statements (loops)-while, for, do-while statements, Loop examples, Nested loops, break, continue, go to statement. ***(Text Book-1: Chapter 2 & Chapter 3)***	
UNIT – II	**05 Hours**
Arrays: Introduction, declaration & initialization of array, reading and writing array elements,	

COURSE SYLLABUS

SCHOOL OF ENGINEERING

Operations on array: Traversal, searching (Linear and Binary search), sorting (Bubble sort and Selection Sort). Declaration and Initialization of two-dimensional arrays. Matrix Operations (addition, subtraction, multiplication, transpose) using two-dimensional array. **Strings:** Definition, declaration, initialization, and representation. String handling functions and character handling functions. ***(Text Book-1: Chapter 5:5.1 to 5.9 & Chapter 6)***	
UNIT – III	**06 Hours**
Pointers: Definition and declaration and initialization of pointers. Accessing values using pointers. Accessing array elements using pointers. **Functions:** Definition and declaration. Built-in functions and User-defined functions. Categories of functions with example. Pointers as function arguments, array as function argument, Call-by-value and call-by-reference. Recursion. ***(Text Book-1: Chapter 7: 7.1 to 7.17 & Chapter 4:4.1 to 4.8, 4.10)***	

COURSE SYLLABUS

SCHOOL OF ENGINEERING

UNIT – IV	04 Hours
Structures: Purpose and usage of structures. Declaration of structures. Assignment with structures. Structure variables and arrays. Nested structures. Student and employee database implementation using structures. **Unions**: Declaration and initialization of a union. Difference between structures and unions. Example programs. ***(Text Book-1: Chapter 8: 8.1, 8.2,8.6)***	
UNIT – V	**04 Hours**
Memory allocation in C programs: Dynamic memory allocation, memory allocation process, allocating a block of memory, releasing the used space, altering the size of allocated memory. **Files**: Defining, open, read, write, seek and closing of both textual and random files. ***(Text Book-1: Chapter 7: 7.18 to 7.20 & Chapter 9: 9.1 to 9.5, 9.8)***	

PRESCRIBED BOOKS

SCHOOL OF ENGINEERING

TEXT BOOKS:

1. Reema Thareja, "Programming in C". Oxford University Press, Second Edition, 2016.

REFERENCE BOOKS:

1. Brian W. Kernigham and Dennis M. Ritchie, (2012) "The C Programming Language", 2nd Edition, PHI.
2. Behrouz A. Forouzan, Richard F. Gilberg, "Computer Science - A Structured Approach Using C", Cengage Learning, 2007.
3. Vikas Gupta, "Computer Concepts and C Programming", Dreamtech Press 2013.

COURSE OUTCOMES

Dayananda Sagar University Bengaluru

COURSE OUTCOME	DESCRIPTION	Blooms Taxonomy Level
At the end of the Course the Student will be able to:		
1	**Apply** fundamental programming constructs such as data types, operators, control flow, and loops to develop basic C programs.	L2, L3
2	**Demonstrate** the ability to use one-dimensional and two-dimensional arrays, perform matrix operations, and manipulate strings using standard C functions.	L3
3	**Apply** pointer concepts and function mechanisms including recursion, call-by-value/reference, and arrays as arguments to build modular C programs.	L3
4	**Develop** programs using structures and unions to represent and manage compound data types for real-world applications like employee or student databases.	L3
5	**Implement** dynamic memory allocation techniques and file operations in C for efficient data storage and retrieval.	L3

PROGRAM ATTRIBUTES

SCHOOL OF ENGINEERING

- Engineering knowledge
- Problem analysis
- Design/development of solutions
- Investigation of complex problems
- Modern tool usage
- Ethics and professional responsibility
- Communication and teamwork
- Environment and sustainability
- Lifelong learning
- Project management and finance
- Societal and global impact

PSO1: Design and Integrate software and hardware systems by following standard software engineering principles in the areas related to IOT, Cloud, Networks, Security, Embedded Systems, and Artificial Intelligence of varying complexity.

PSO2: Design and Implement application software systems by applying the concepts of Programming languages, Machine Learning, Mobile Computing, and Data Science that meet the automation requirements of society and Industry.

CO-PO MATRIX

SCHOOL OF ENGINEERING

MAPPING LEVELS OF COs to POs/PSOs													
	PROGRAM OUTCOMES (PO)											PSO	
CO	1	2	3	4	5	6	7	8	9	10	11	1	2
CO1	3											1	
CO2		2										1	
CO3			2									1	
CO4												1	
CO5					2							1	

CIA and SEE MARKS DISTRIBUTION

SCHOOL OF ENGINEERING

Evaluation Components	Weightage	
Mid Semester Examination	**30**	60 Marks
Alternate Assessment Tool (AAT)	**10**	
Lab Record	**10**	
Lab Examination	**10 [4W+4E+2V]**	
Semester End Examination (SEE)		40 Marks
	Total: 100 Marks	

*__Alternate Assessment Tool__ (AAT) includes *Project based learning/E-course certification/Case study.*

Advantages Of C Over Python language

SCHOOL OF ENGINEERING

1. Performance

C is a compiled language, which means it is **converted directly into machine code that the processor can execute**. This often **results in faster execution times** compared to Python, which is an interpreted language.

2. Memory Management

C provides **more control over memory management**. You **can allocate and deallocate memory manually** using functions like malloc() and free(). Useful where memory usage is critical.

Advantages Of C Over Python language

SCHOOL OF ENGINEERING

3. Low-Level Programming

C is closer to the hardware, making it **ideal for system-level programming** such as **writing operating systems, drivers, and embedded systems**. It allows direct manipulation of hardware and memory addresses.

4. Deterministic Resource Management

In C, you have **deterministic control over resource management**, which is **essential for real-time systems** where timing is critical. Python's garbage collection can introduce unpredictability in resource management.

Advantages Of C Over Python language

SCHOOL OF ENGINEERING

5. Smaller Executable Size

Programs written in C generally **produce smaller executable files** compared to Python, which can be beneficial for applications with **limited storage space.**

6. Learning Foundation

Learning C provides **a strong foundation in programming concepts** such as **pointers, memory allocation, and low-level operations**, which can be beneficial for understanding more complex languages and systems.

C Language or Python language ??

SCHOOL OF ENGINEERING

- Both **C** and **Python** have their **own strengths** and are **suited for different types of tasks.**

- The **choice between C and Python** often **depends on the specific requirements of the project.**

NEED OF LANGUAGES

Why Need of Languages ??

SCHOOL OF ENGINEERING

- Importance of regional languages for human beings.
- Importance of computer languages for human beings.

Types of Computer Languages

SCHOOL OF ENGINEERING

- Low Level Language
- Middle Level Language
- High Level Language

1. Machine Level Language

SCHOOL OF ENGINEERING

Machine Level Language: This is also called as the **First-Generation Computer Languages**. The Machine Language Programs **contains all the instructions in the Binary Code** Fashion.

Advantages of Machine Language

- It makes **fast and efficient** use of the computer.
- It **requires no translator** to translate the code i.e. Directly understood by the computer.

Disadvantages of Machine Language:

- All **operation codes have to be remembered**.
- All **memory addresses have to be remembered**.
- It is **hard to amend or find errors** in a program written

2. Assembly Level Language

SCHOOL OF ENGINEERING

Assembly Language: It is **Second Generation Language. T**here are **Many Mnemonics** which have Some Specific Meaning.

Advantages of Assembly Language

- It is **easier to understand and use** as compared to machine language.
- It is **easy to locate and correct errors**.

Disadvantages of Assembly Language

- Like machine language it is also **machine dependent**.
- The programmer should **have the knowledge of the hardware** also.

3. High Level Language

SCHOOL OF ENGINEERING

High Level Language:

All the High-Level Programming Languages are User Friendly.

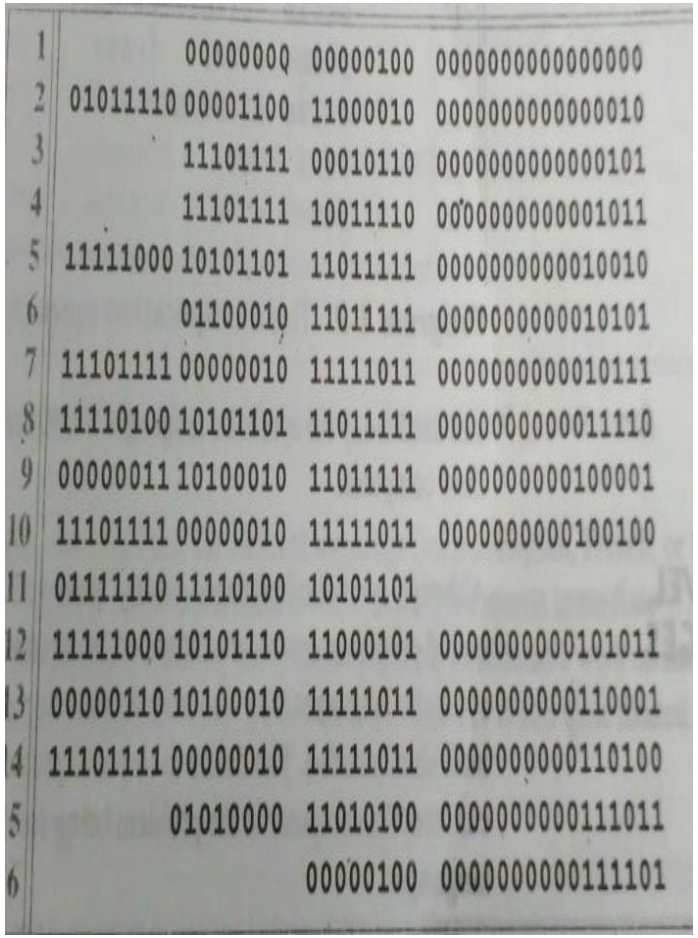
It means the Syntax of theses Languages are Quite Simple and are in the Form of English Language.

Advantages of High-Level Language

1. User-friendly
2. Similar to English with vocabulary of words and symbols
3. Easier to learn.
4. They require less time to write.
5. They are easier to maintain.
6. Problem oriented rather than 'machine' based.

SCHOOL OF ENGINEERING

Machine vs Assembly vs High-Level Language



```
1        00000000 00000100 0000000000000000
2 01011110 00001100 11000010 0000000000000010
3        11101111 00010110 0000000000000101
4        11101111 10011110 0000000000001011
5 11111000 10101101 11011111 0000000000010010
6        01100010 11011111 0000000000010101
7 11101111 00000010 11111011 0000000000010111
8 11110100 10101101 11011111 0000000000011110
9 00000011 10100010 11011111 0000000000100001
10 11101111 00000010 11111011 0000000000100100
11 01111110 11110100 10101101
12 11111000 10101110 11000101 0000000000101011
13 00000110 10100010 11111011 0000000000110001
14 11101111 00000010 11111011 0000000000110100
15       01010000 11010100 0000000000111011
16               00000100 0000000000111101
```

```
. Model Small
. Data
A DB ?
...
...
. CODE
   Mov Ax , 08
   Mov Bx, 10
   Add Ax, Bx
END
```

```
#include <stdio.h>

int main (void)
{
/* Local Definitions */
   int number1;
   int number2;
   int result;

/* Statements */
   scanf  ("%d", &number1);
   scanf  ("%d", &number2);
   result = number1 * number2;
   printf ("%d", result);
   return 0;
} /* main */
```

3. High Level Language

SCHOOL OF ENGINEERING

High Level Language:

All the High-Level Programming Languages are User Friendly.

It means the Syntax of theses Languages are Quite Simple and are in the Form of English Language.

Advantages of High-Level Language

1. User-friendly
2. Similar to English with vocabulary of words and symbols
3. Easier to learn.
4. They require less time to write.
5. They are easier to maintain.
6. Problem oriented rather than 'machine' based.

Let's C

What is C Language ??

SCHOOL OF ENGINEERING

- **C is a general-purpose programming language** developed by **Dennis Ritchie** in the early **1970s at AT&T Bell Labs**.
- It was designed for **system programming**, particularly for writing operating systems.
- **The UNIX operating system** was one of the first major programs written in C.

Why Learn C Language ??

SCHOOL OF ENGINEERING

- **Simplicity:** C is known for its **simple syntax**, which makes it a great language for beginners.
- **Efficiency:** Programs written in C are **efficient and fast**.
- **Portability:** C code is portable, meaning it can **run on different types of machines** with minimal changes.
- **Low-level access:** C **provides access to low-level memory through pointers**, which allows for efficient programming

Where to apply C Language ??

SCHOOL OF ENGINEERING

- **Operating Systems:** The UNIX operating system is written in C.
- **Embedded Systems:** Due to its efficiency, C is used in programming microcontrollers and embedded systems.
- **Game Development:** Some game engines use C for high-performance tasks.
- **Compilers :** Many compilers, interpreters and drivers are written in C.
- **Databases and Word processors:** Many softwares are written in C

Conclusion

SCHOOL OF ENGINEERING

- Learning C programming language can provide **a strong foundation in programming principles** and concepts.
- Can understand **how software interacts with hardware.**
- Many modern languages (like C++, Java, and Python) **have inherited some concepts from C**.

SCHOOL OF ENGINEERING

THANK YOU ALL

Let us put in the Best Efforts