B. Sc. Part – I Semester – II

COMPUTER SCIENCE (Optional)

Programming Skills Using 'C'

Question Bank

1. **Functions:**
   - **What is a function in C, and why is it useful?**
   - **How do you declare and define functions in C?**
   - **Can you explain the concepts of pass by value and pass by reference in function arguments?**

   1. **Function Basics:**
      - **What is a function in C, and why is it useful?**
      - **How do you declare and define a function in C?**
      - **Can you explain the difference between function declaration and definition?**
   2. **Function Parameters:**
      - **What are function parameters, and how are they used?**
      - **How do you pass arguments to a function in C?**
      - **Can you describe the difference between pass by value and pass by reference?**
   3. **Return Values:**
      - **How do you specify the return type of a function in C?**
      - **What happens when a function returns a value in C?**
      - **Can you explain the significance of the void return type in C functions?**
   4. **Function Prototypes:**
      - **What is a function prototype, and why is it needed?**
      - **How do you declare a function prototype in C?**
      - **What are the advantages of using function prototypes?**
   5. **Recursive Functions:**
      - **What is recursion, and how does it work in C functions?**
      - **Can you provide an example of a recursive function in C?**
      - **What are the advantages and disadvantages of using recursion?**
   6. **Function Pointers:**
      - **What is a function pointer, and how is it declared in C?**
      - **How do you use function pointers to pass functions as arguments or return values?**
      - **Can you provide an example of using function pointers in a practical scenario?**
   7. **Library Functions:**
      - **What are library functions, and how do you use them in C?**
      - **Can you provide examples of commonly used library functions in C, such as printf(), scanf(), or strlen()?**
      - **How do you include external libraries and use their functions in your C programs?**

2. **Arrays and Strings:**
   - **What is an array in C, and how do you declare and initialize one?**
   - **Explain how to access individual elements of an array in C.**
   - **What are strings in C, and how are they different from arrays?**

Array Basics:
What is an array in C? How is it different from a single variable?
How do you declare and initialize an array in C?
Can you explain the concept of array indexing and how it works?
Array Operations:
How do you access individual elements of an array?

What is the significance of the array size in C? How do you determine the size of an array?
How do you iterate through an array using loops?
Multidimensional Arrays:
What are multidimensional arrays in C? How do you declare and initialize them?
Can you explain the concept of row-major order in multidimensional arrays?
How do you access elements of a multidimensional array?
String Basics:
What is a string in C? How is it represented?
How do you declare and initialize a string in C?
What is the null character ('\0') and why is it important in C strings?
String Operations:
How do you calculate the length of a string in C?
What are some common string manipulation functions provided by the C standard library?
How do you compare two strings in C?
Character Arrays vs. Strings:
What is the difference between a character array and a string in C?
How do you convert between character arrays and strings?
Can you explain the importance of null-terminated strings in C?
String Input and Output:
How do you read a string from the user in C?
How do you print a string to the console in C?
What precautions should be taken when dealing with input and output of strings to avoid buffer overflow or truncation?

3. **Pointers:**
   - **What is a pointer in C, and why is it used?**
   - **How do you declare, initialize, and dereference pointers in C?**
   - **Provide examples of pointer arithmetic and its applications.**

Basics of Pointers:
What is a pointer in C? How does it differ from regular variables?
Can you explain the concept of memory addresses and how pointers are used to access them?
How do you declare and initialize a pointer variable in C?
Pointer Arithmetic:
What is pointer arithmetic, and how is it performed in C?
Can you explain the difference between incrementing a pointer and dereferencing it?
How does pointer arithmetic relate to array indexing in C?
Pointer and Arrays:
How are pointers and arrays related in C?
Can you explain how array names are essentially pointers to the first element of the array?
What are the advantages of using pointers for array manipulation?
Pointer and Functions:
How are pointers used as function parameters in C?
Can you explain the concept of pass-by-reference using pointers?
Provide an example of a function that swaps two variables using pointers.
Dynamic Memory Allocation:
What is dynamic memory allocation, and why is it necessary?
How do you allocate and deallocate memory dynamically using functions like `malloc()` and `free()`?

What precautions should be taken when using dynamic memory allocation to avoid memory leaks or undefined behavior?

Pointer to Pointers:

What is a pointer to a pointer, and when would you use it?

Provide an example of using a pointer to a pointer to implement a two-dimensional array dynamically.

How does a pointer to a pointer differ from a regular pointer?

Pointers and Strings:

How are strings represented in C, and how are they related to pointers?

Explain the difference between using character arrays and pointers to strings.

Provide examples of common string manipulation tasks using pointers.

4. **Memory Management:**
   - **Explain the concepts of stack and heap memory in C.**
   - **How do you allocate and deallocate memory dynamically using malloc, calloc, realloc, and free functions?**
   - **What are memory leaks, and how do you prevent them in C programs?**
5. **Structures and Unions:**
   - **What is a structure in C, and how is it defined?**
   - **How do you declare and access members of a structure in C?**
   - **What are unions, and how are they different from structures?**

Basics of Structures:

What is a structure in C? How does it differ from a basic data type?

Can you explain how to declare and initialize a structure in C?

How do you access individual members of a structure?

Structures and Functions:

How can structures be passed to functions in C?

Can you write a function that takes a structure as an argument and modifies its members?

How do you return a structure from a function?

Nested Structures:

What is a nested structure, and why might you use one?

How do you declare and access members of a nested structure?

Can you give an example of when you might use a nested structure?

Unions:

What is a union in C, and how does it differ from a structure?

How do you declare and access members of a union?

Can you explain the concept of memory sharing in unions?

Comparison Between Structures and Unions:

What are the main differences between structures and unions?

When would you choose to use a structure over a union, and vice versa?

Can you provide examples of scenarios where structures or unions would be more appropriate?

Practical Applications:

How can structures and unions be used to represent real-world entities or complex data?

Can you give examples of how structures and unions are used in programming libraries or APIs?

How might you use structures and unions to efficiently manage and manipulate data in a large-scale program?

6. **File Handling:**
   - **How do you open, read from, write to, and close files in C?**
   - **Explain the concepts of file modes and file pointers in C.**
   - **Provide an example of reading and writing data to a text file in C.**

File Operations:
How do you open a file in C for reading and writing?
What are the differences between reading and writing modes when opening a file?
Can you explain the purpose of the `fopen()` function and its parameters?

Reading from Files:
How do you read characters from a file in C using standard I/O functions?
What is the significance of the `feof()` function when reading from a file?
Can you describe the difference between reading characters using `fgetc()` and `fgets()`?

Writing to Files:
How do you write data to a file in C using standard I/O functions?
What are the differences between using `fputc()` and `fputs()` for writing characters to a file?
How do you handle errors while writing to a file in C?

File Positioning:
How do you move the file pointer to a specific position in a file in C?
What is the significance of the `fseek()` function, and how is it used?
Can you explain the concept of file positioning modes (`SEEK_SET`, `SEEK_CUR`, `SEEK_END`)?

Binary File Handling:
What are binary files, and how are they different from text files?
How do you read and write binary data to and from a file in C?
Can you discuss the importance of using binary mode (`"rb"`, `"wb"`, `"ab"`) when working with binary files?

Error Handling and File Closing:
How do you check for errors when performing file operations in C?
What is the significance of the `fclose()` function, and when should it be called?
How do you ensure proper error handling and file closing in a C program that deals with file operations?

Advanced File Handling:
Can you discuss how file handling functions like `fprintf()` and `fscanf()` are used for formatted input/output?
How do you read and write records of a specific structure from and to a file?
What are some advanced techniques for error handling and robust file handling in C?