DEPARTMENT OF COMPUTER SCIENCE,
RABINDRANATH TAGORE UNIVERSITY, HOJAI: ASSAM

Computer Science Syllabus for UG courses, according to NEP2020 (CBCS)

COMPUTER FUNDAMENTALS AND PROGRAMMING

MIN-CSC-3.2

Class	Courses	Subject papers	Credits
SEM-III	CORE	COMPUTER FUNDAMENTALS AND PROGRAMMING	3
		Practical	1
			Total credit:4

Note: *COR - Core;*

Note: Students who exits after first year will be awarded undergraduate certificate (in the field/discipline) if He/She completes a summer internship course of credit-4, in Second year.

SEMESTER-III
CORE COURSE
MIN – CSC – 3.2 (COMPUTER FUNDAMENTALS AND PROGRAMMING)

Total marks: 100 -- (Theory: 45, Internal Assessment: 30, Practical 25) Total credit: 4 (L-3, P-1, T-4)

1. ## Course objectives:
- At the end of the course, students will be able to:
- Understanding the basics of Computer and programing
- Adopt algorithmic approach to solve problems using pseudocode and flowcharts
- Understanding and write programs in C to implement conditions, loops, functions and other
- Programming constructs
- Work on arrays, strings and basic file operations in C

2. **Perquisite** : NIL
3. **Semester**: 3
4. **Course Level** : 100 – 199
5. **Theory Credit** :3
6. **Practical Credit** :1
7. **Number of required hours** :
   a) **Theory** : 45 hours
   b) **Practical** : 30 hours
   c) **Non-Contact** : 5 hours
8. **List of reference books**:
a) B.S. Gottfried, "Schism's Outline of Theory and Problems of Programming with C", McGraw Hill, 2007.
b) B. Kernighan, Ritchie," The C Programming Language:, Second Edition, Prentice Hall 1958
c) E Balaguruswami, "Programming in ANSI C" , 2nd Ed., Tata McGraw Hill, 2004
d) V. Rajaraman, "Fundamentals of computer" , 4th Ed .,PHI, 2006
e) R. Thareja, "Computer Fundamentals & Programming in C ", Oxford University Press, 2013.

Detailed Syllabus:
UNIT 1: Computer Fundamentals   (Lectures: 09)

Introduction to computer hardware, software – application and system software. Operating systems,. Major components of a Digital Computer – ALU and CU, Memory – primary and secondary memory. Storage devices – magnetic storage devices, optical storage devices, Input devices – mouse, keyboard, touch –screen, scanner etc.

Output devices – CRT/LCD/ LED monitors, printers etc. Number systems – binary, octal, hexadecimal, BCD. Conversion between two number systems. Signed magnitude, 1's complement and 2's complement representation. Character encoding – ASCII, EBCDIC, Unicode. Basic overview of networks and the Internet, WWW

UNIT 2: Programming Basics   **(Lectures: 04)**

Introduction to programming languages. Low-level and high-level language and their characteristics. Compiler vs. interpreter IDE. Bugs and its types. Algorithms, pseudocodes and flowcharts. Overview of the C Programming language. Structure of a C Program.

UNIT 3: Datatypes and Operators: **(Lectures: 05)**

Basic data types in C – integers, floats, doubles, characters, and void. Size and range of values of

Data types. Variables, declaring variables. Operators and expressions. Input and output statements getchar (), getc (), putchar (), putc (), puts (), scanf (), printf (), format specifiers, Typecasting, Operators in C - binary and unary operators. Arithmetic, assignment, logical, comparison, bitwise and conditional operators. Order of precedence of operator. Associativity of operators. Expressions and statements in C. L- Value and R- Value. Basic syntax and semantics for expressions and statements.

UNIT 4: Control Structures and Functions: **(Lectures: 08)**

Control structures in C, decision making with if, if-else, switch statements, Nested conditions, looping with while, do-while, and for statement. Break and continue statements. Nested loops, Introduction to functions. Function prototypes and arguments. Defining and calling functions in c. return values and types. Formal and actual Parameter. Call by value. Call by reference, Introduction to recursion. Writing recursive functions in C. Importance of main () function, return type of main () function.

UNIT 5: Arrays and Strings   (**Lectures: 05**)

Introduction to Arrays. Declaration and initialization of arrays, accessing array elements. Multidimensional arrays, Introduction to strings. Strings String input and output in C. String manipulation functions in C – strlen (), strcpy (), strcat (), strcmp ()

UNIT 6: Pointers and Memory allocation   (**Lecture: 06**)

Introduction to pointers, Pointer declaration and initialization. Pointers and addresses, Pointers and arrays, Pointers and functions, Review of call by reference, pointer arithmetic, passing an array using pointer in function cell. Introduction to dynamic memory allocation. Allocation and deallocation of memory using malloc (), calloc (), and free () functions.

UNIT 7: STRUCTURE AND UNION   (**Lecture: 04**)

Introduction to structures Declaration and initialization of structures. Accessing structure members. Nested structures and arrays of structures, Unions in Declaration and initialization of unions. Accessing union members, Differences between structures and uni8ons .Typedef.

UNIT 8: FILE HANDLING AND PREPROCESSOR DIRECTIVES   **(Lecture: 04)**

      Introduction to file handling in C. Opening and closing files – fopen () fclose (). Modes of opening a file. Binary files and text files. Reading and writing files –fgetc (), fgets (), fread () fputc () fputs () fwrite (). File pointers. Error handling I file operations, Preprocessor directives in C #define, #include, #ifdef, #ifndef, and #endif directives using preprocessor directives to define constants and macros. Header files.

Lists of Practical

(This is a suggestive lists only .Questions need not be restricted to this list. The practical are advised to be performed in Linux environment.)

1. Write a program in C to print "Hello World."
2. Write a program to take input of two numbers and print their sum, product, difference.
3. Write a program to find the smallest or greatest of three numbers given as input.
4. Write a program to print the sum and product of digits of an integer.
5. Write a program to print a triangle of stars as follows (take number of lines from user):

```
        *
       * * *
      * * * * *
     * * * * * * *
    * * * * * * * * *
```

6. Write a program to reverse a number.
7. Write a program to compute the sum of the first n terms of the following series S= 1+1/2+1/3+1/4+……
8. Write a program to compute the sum of the first n terms of the following series S= 1-2+3-4+5………
9. Write a function that checks whether a given string is palindrome or not. Use this function to find whether
    the string entered by user is palindrome or not.
10. Write a function to find whether a given number is prime or not. Use the same to generate the prime
    numbers less than 100.
11. Write a program to compute the factors of a given number.
12. Write a program to display Fibonacci series (i) using recursion, (ii) using iteration.
13. Write a program to calculate Factorial of a number (i) using recursion, (ii) using iteration
14. Write a program in which a function is passed address of two variables and then alter its contents.
15. Write a program which takes the radius of a circle as input from the user, passes it to another function
    that computes the area and the circumference of the circle and displays the value of area and circumference
    From the main () function.
16. Write a program to create an array with inputs from the user and print the same.
17. Write a program to perform following actions on an array entered by the user:
    a) Print the even-valued elements
    b) Print the odd-valued elements
    c) Calculate and print the sum and average of the elements of array
    d) Print the maximum and minimum element of array
    e) Remove the duplicates from array
    f) Print the array in reverse order
       the program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.

18. Write a program to take a matrix from the user and print the transpose of the same.
19. Write a program to take two matrices from the user and find the sum and product of both.
20. Write a program to perform following operations on strings:
      a) Convert all lowercase characters to uppercase
      b) Convert all uppercase characters to lowercase
      c) Calculate number of vowels in the strings
      d) Reverse the string
      e) Concatenate two strings without using strcat() function
      f) Concatenate two strings using strcat() function
      g) Compare two strings using strcmp()
      h) Copy one string to another using strcpy()

****

DEPARTMENT OF COMPUTER SCIENCE, RABINDRANATH TAGORE UNIVERSITY, HOJAI: ASSAM

Computer Science Syllabus for UG courses, according to NEP2020 (CBCS)

Subject details for Fourth Semester: Paper: 4.2

Class	Courses	Subject papers	Credits
SEM-IV	CORE	COMPUTER FUNDAMENTALS AND PROGRAMMING	3
		Practical	1
			Total credit:4

Note: *COR - Core;*

Note: Students who exits after first year will be awarded undergraduate certificate (in the field/discipline) if He/She completes a summer internship course of credit-4, in Second year.

MIN – CSC – 4.2 (COMPUTER FUNDAMENTALS AND PROGRAMMING)

Total marks: 100 -- (Theory: 45, Internal Assessment: 30, Practical 25) Total credit: 4 (L-3, P-1, T-4)

1. ## Course objectives:
- At the end of the course, students will be able to:
- Understanding the basics of Computer and programing
- Adopt algorithmic approach to solve problems using pseudocode and flowcharts
- Understanding and write programs in C to implement conditions, loops, functions and other
- Programming constructs
- Work on arrays, strings and basic file operations in C
2. **Perquisite** : NIL
3. **Semester**: 3
4. **Course Level** : 100 – 199
5. **Theory Credit** :3
6. **Practical Credit** :1
7. **Number of required hours** :
   a) **Theory** : 45 hours
   b) **Practical** : 30 hours
   c) **Non-Contact** : 5 hours
8. **List of reference books**:
a) B.S. Gottfried, "Schism's Outline of Theory and Problems of Programming with C", McGraw Hill, 2007.
b) B. Kernighan, Ritchie," The C Programming Language:, Second Edition, Prentice Hall 1958
c) E Balaguruswami, "Programming in ANSI C" , 2nd Ed., Tata McGraw Hill, 2004
d) V. Rajaraman, "Fundamentals of computer" , 4th Ed .,PHI, 2006
e) R. Thareja, "Computer Fundamentals & Programming in C ", Oxford University Press, 2013.

Detailed Syllabus:
UNIT 1: Computer Fundamentals   **(Lectures: 09)**

Introduction to computer hardware, software – application and system software. Operating systems,. Major components of a Digital Computer – ALU and CU, Memory – primary and secondary memory. Storage devices – magnetic storage devices, optical storage devices, Input devices – mouse, keyboard, touch –screen, scanner etc.

Output devices – CRT/LCD/ LED monitors, printers etc. Number systems – binary, octal, hexadecimal, BCD. Conversion between two number systems. Signed magnitude, 1's complement and 2's complement representation. Character encoding – ASCII, EBCDIC, Unicode. Basic overview of networks and the Internet, WWW

UNIT 2: Programming Basics   (Lectures: 04)

Introduction to programming languages. Low-level and high-level language and their characteristics. Compiler vs. interpreter IDE. Bugs and its types. Algorithms, pseudocodes and flowcharts. Overview of the C Programming language. Structure of a C Program.

UNIT 3: Datatypes and Operators: (Lectures: 05)

Basic data types in C – integers, floats, doubles, characters, and void. Size and range of values of

Data types. Variables, declaring variables. Operators and expressions. Input and output statements getchar (), getc (),   putchar (), putc (), puts (), scanf (), printf (), format specifiers, Typecasting, Operators in C - binary and unary operators. Arithmetic, assignment, logical, comparison, bitwise and conditional operators. Order of precedence of operator. Associativity of operators. Expressions and statements in C. L- Value and R- Value. Basic syntax and semantics for expressions and statements.

UNIT 4: Control Structures and Functions: (Lectures: 08)

Control structures in C, decision making with if, if-else, switch statements, Nested conditions, looping with while, do-while, and for statement. Break and continue statements. Nested loops, Introduction to functions. Function prototypes and arguments. Defining and calling functions in c. return values and types. Formal and actual Parameter. Call by value. Call by reference, Introduction to recursion. Writing recursive functions in C. Importance of main () function, return type of main () function.

UNIT 5: Arrays and Strings   (Lectures: 05)

Introduction to Arrays. Declaration and initialization of arrays, accessing array elements. Multidimensional arrays, Introduction to strings. Strings String input and output in C. String manipulation functions in C – strlen (), strcpy (), strcat (), strcmp ()

UNIT 6: Pointers and Memory allocation   (Lecture: 06)

Introduction to pointers, Pointer declaration and initialization. Pointers and addresses, Pointers and arrays, Pointers and functions, Review of call by reference, pointer arithmetic, passing an array using pointer in function cell. Introduction to dynamic memory allocation. Allocation and deallocation of memory using malloc (), calloc (), and free () functions.

UNIT 7: STRUCTURE AND UNION   (Lecture: 04)

Introduction to structures Declaration and initialization of structures. Accessing structure members. Nested structures and arrays of structures, Unions in Declaration and initialization of unions. Accessing union members, Differences between structures and uni8ons .Typedef.

UNIT 8: FILE HANDLING AND PREPROCESSOR DIRECTIVES   (Lecture: 04)

Introduction to file handling in C. Opening and closing files – fopen () fclose (). Modes of opening a file. Binary files and text files. Reading and writing files –fgetc (), fgets (), fread () fputc () fputs () fwrite (). File pointers. Error handling I file operations, Preprocessor directives in C #define, #include, #ifdef, #ifndef, and #endif directives using preprocessor directives to define constants and macros. Header files.

Lists of Practical

(This is a suggestive lists only .Questions need not be restricted to this list. The practical are advised to be performed in Linux environment.)

1. Write a program in C to print "Hello World."
2. Write a program to take input of two numbers and print their sum, product, difference.
3. Write a program to find the smallest or greatest of three numbers given as input.
4. Write a program to print the sum and product of digits of an integer.
5. Write a program to print a triangle of stars as follows (take number of lines from user):

```
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

6. Write a program to reverse a number.
7. Write a program to compute the sum of the first n terms of the following series S= 1+1/2+1/3+1/4+……
8. Write a program to compute the sum of the first n terms of the following series S= 1-2+3-4+5………
9. Write a function that checks whether a given string is palindrome or not. Use this function to find whether
   the string entered by user is palindrome or not.
10. Write a function to find whether a given number is prime or not. Use the same to generate the prime
    numbers less than 100.
11. Write a program to compute the factors of a given number.
12. Write a program to display Fibonacci series (i) using recursion, (ii) using iteration.
13. Write a program to calculate Factorial of a number (i) using recursion, (ii) using iteration
14. Write a program in which a function is passed address of two variables and then alter its contents.
15. Write a program which takes the radius of a circle as input from the user, passes it to another function
    that computes the area and the circumference of the circle and displays the value of area and circumference
    From the main () function.
16. Write a program to create an array with inputs from the user and print the same.
17. Write a program to perform following actions on an array entered by the user:
    a) Print the even-valued elements
    b) Print the odd-valued elements
    c) Calculate and print the sum and average of the elements of array
    d) Print the maximum and minimum element of array
    e) Remove the duplicates from array
    f) Print the array in reverse order
       the program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
18. Write a program to take a matrix from the user and print the transpose of the same.
19. Write a program to take two matrices from the user and find the sum and product of both.
20. Write a program to perform following operations on strings:
    a) Convert all lowercase characters to uppercase
    b) Convert all uppercase characters to lowercase
    c) Calculate number of vowels in the strings
    d) Reverse the string
    e) Concatenate two strings without using strcat() function
    f) Concatenate two strings using strcat() function
    g) Compare two strings using strcmp()
    h) Copy one string to another using strcpy()

****