GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)
Competency-focused Outcome-based Green Curriculum-2021 (COGC-2021)
I – Semester

Course Title: **Basic Computer Programming**
(Course Code: 4310702)

Diploma programme in which this course is offered	Semester in which offered
Computer Engineering	First

1. RATIONALE

The present era can be said a digital era. Nowadays almost in every walk of life there is application of digitization, atomization as well as connecting various gadgets, home appliances, human body etc. to each other. The core component which drives these tasks is a piece of code for the machine, known as a program. It is essential for the students to learn basic concepts and methodology to develop computer programs.

This first and introductory level Computer Programming Course is intended to develop logical thinking skills and programs using a popular structured programming language `C'. The programming skills thus acquired can be used for developing programs for the scientific, research and business purposes.

2. COMPETENCY

The purpose of this course is to help the student to attain the following industry identified competency through various teaching learning experiences.

.

- **Develop structured, modular and memory efficient programs in 'C'.**

3. COURSE OUTCOMES (COs)

The practical exercises, the underpinning knowledge and the relevant soft skills associated with this competency are to be developed in the student to display the following COs:

a. Design algorithm and flowchart for the given Problem.
b. Develop C programs using control structures.
c. Develop C programs using arrays and pointers.
d. Implement user defined functions.
e. Use structure and union in C programs.
f. Implement file and I/O operations in C.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme (In Hours)			Total Credits (CI+T/2+P/2)	Examination Scheme				
				Theory Marks		Practical Marks		Total Marks
L	T	P	C	CA	ESE	CA	ESE	
3	0	4	5	30*	70	25	25	150

(*): Out of 30 marks under the theory CA, 10 marks are for assessment of the micro-project to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for assessing the attainment of the cognitive domain UOs required for the attainment of the COs.

Legends: L-*Lecture;* ***T*** *– Tutorial/Teacher Guided Theory Practice;* ***P*** *- Practical;* ***C*** *– Credit,* ***CA*** *- Continuous Assessment;* ***ESE*** *- End Semester Examination.*

5. SUGGESTED PRACTICAL EXERCISES

The following practical outcomes (PrOs) are the sub-components of the COs. *These PrOs need to be attained to achieve the Cos.*

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. required
1	Practice using Visual Programming Language like scratch	I	02
2	Design and test sample C programs to display a message on screen.	II	01
3	Design and test minimum 3 C programs using constants, variables and datatypes.	II	02
4	Design and test a C program to swap 2 numbers using a third variable and without using a third variable.	II	01
5	Design and test a C program to compute volume and surface area of a sphere.	II	01
6	Design and test a C program to convert temperature in Fahrenheit to Celsius and vice versa.	II	01
7	Design and test at least 4 C programs to using enlisted operators: (1) Assignment (2) Arithmetic (3) Relational (4) Logical	II	02
8	Design and test at least 5 C programs using the enlisted operators: (1) Bitwise (2) Increment and Decrement (3) Conditional (4) Comma (5) size of	II	02
9	Design and test at least 3 C programs to test the operator precedence and their associativity, implicit and explicit type conversion.	II	02
10	Design and test at least 3 C programs to show formatted and unformatted input and output.	II	02
11	Design and test at least 2 C programs using decision making statements: (1) Simple if (2) if…else (3) Nested if (4) if…else ladder (5) switch (6) goto	III	03
12	Design and test at least 3 C programs using the for loop.	III	02

13	Design and test at least 3 C programs using the while loop.	III	02
14	Design and test at least 3 C programs using do…while loop.	III	02
15	Design and test a C program using break and continue statements.	III	01
16	Design and test at least 5 pattern programs using loop structures.	III	03
17	Design and test at least 5 C programs using one dimensional array.	IV	02
18	Design and test at least 3 C programs using two dimensional arrays.	IV	02
19	Design and test at least 3 C programs using strings.	IV	02
20	Design and test at least 3 C programs using pointers.	IV	02
21	Design and test a C program using the concept of pointer to pointer.	IV	01
22	Design and test at least 5 C programs using user defined functions.	V	04
23	Design and test at least 3 C programs by applying the recursion concept.	V	02
24	Design and test a C program to test various inbuilt string functions.	V	02
25	Design and test a C program to demonstrate various inbuilt math functions.	V	02
26	Design and test a C program to demonstrate storage classes.	V	02
27	Design and test a C program to demonstrate usage of enum and typedef.	VI	02
28	Design and test at least 3 C programs on structures and unions.	VI	02
29	Design and test at least 2 C programs using file operations.	VI	02
	Total		**56**

Note

i. *More **Practical Exercises** can be designed and offered by the respective course teacher to develop the industry relevant skills/outcomes to match the COs. The above table is only a suggestive list.*

ii. *The following are some **sample** 'Process' and 'Product' related skills (more may be added/deleted depending on the course) that occur in the above listed **Practical Exercises** of this course required which are embedded in the COs and ultimately the competency..*

S. No.	Sample Performance Indicators for the PrOs	Weightage in %
1	Correctness of algorithm/program	30
2	Readability and documentation of the program/Quality of input and output displayed (messaging and formatting)	10
3	Code efficiency	20
4	Debugging ability	20
5	Program execution/answer to sample questions	20
Total		**100**

6. MAJOR EQUIPMENT/ INSTRUMENTS AND SOFTWARE REQUIRED

These major equipment with broad specifications for the PrOs is a guide to procure them by the administrators to usher in uniformity of practical in all institutions across the state.

S. No.	Equipment Name with Broad Specifications	PrO. No.
1	Computer with basic configuration with windows or unix os	All
2	C Compiler	All

7. AFFECTIVE DOMAIN OUTCOMES

The following ***sample*** Affective Domain Outcomes (ADOs) are embedded in many of the above mentioned COs and PrOs. More could be added to fulfil the development of this course competency.

a) Follow safety practices.
b) Practice good housekeeping.
c) Demonstrate working as a leader/a team member.
d) Maintain tools and equipment
e) Follow ethical practices.

The ADOs are best developed through the laboratory/field based exercises. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

i. 'Valuing Level' in 1st year
ii. 'Organization Level' in 2nd year.
iii. 'Characterization Level' in 3rd year.

8. UNDERPINNING THEORY

The major underpinning theory is given below based on the higher level UOs of *Revised Bloom's taxonomy* that are formulated for development of the COs and competency. If required, more such UOs could be included by the course teacher to focus on attainment of COs and competency.

Unit	Unit Outcomes (UOs) (4 to 6 UOs at different levels)	Topics and Sub-topics
Unit – I: **Flowchart and Algorithm**	1a. Write pseudo code for the given problem statements 1b. Select appropriate flowchart symbols to represent problem solution graphically 1c. Write algorithms for the given problem statements. 1d. Develop flowchart for the given problem statement 1e. Develop Algorithm for the given problem statement	**Flowchart** 1.1 Definition and Importance of flowchart 1.2 Symbols of flowchart 1.3 Flow lines, Terminals, Input/Output, Processing Decision, Connection off-page connectors 1.4 Guidelines for preparing Flowchart 1.5 Flowchart structure: Sequence, selection, repetition 1.6 Limitation of flowchart **Algorithm** 1.7 Developing and writing algorithm using pseudo codes

Unit	Unit Outcomes (UOs) (4 to 6 UOs at different levels)	Topics and Sub-topics
Unit– II: **Basics of 'C'**	2a. Comprehend general structure of 'C' program 2b. Choose appropriate operators amongst C operators to form expressions in C. 2c. Write simple C programs using arithmetic expressions 2d. Apply different format strings for the input and output using 'C' statements.	**Basics of 'C'** 2.1 General structure of 'C' program and standard directories 2.2 Write, compile, execute a simple 'C' program 2.3 Character set, 'C' tokens 2.4 Keywords and Identifiers 2.5 Data Types in 'C' 2.6 Variables and rules for defining variables, Declaration and Initialization of variables 2.7 Dynamic initialization 2.8 Constant and volatile variable 2.9 Introduction of different types of operators and their symbolic representation, Assignment, Arithmetic, Relational, Logical, Bitwise, Increment and Decrement, Conditional, Comma, size of Operators 2.10 Operator precedence and their associativity 2.11 Evaluation of Expressions 2.12 Type Conversion-Implicit and Explicit 2.13 Input and Output statements in 'C' 2.14 Formatted input and output in 'C'
Unit– III: **Decision Statements and Control Structure**	3a. Develop programs using decision making if-else statement 3b. Develop programs using decision making switch – case statement 3c. Develop programs using unconditional branching goto statements in 'C' language. 3d. Develop C programs using control structure: for, While and Do-While 3e. Apply Break and Continue Statement based on the problem statements in 'C' language.	**Decision Statements** 3.1 Conditional branching statements: Simple if statement 3.2 If-else statement 3.3 Nested If-else statement 3.4 If-else-if Ladder statement 3.5 switch statements 3.6 Unconditional branching statement: goto **Control Statements** 3.7 for loop 3.8 While loop 3.9 Do-while loop 3.10 Nested for loop 3.11 Break and continue statements
Unit–IV:	4a. Develop programs using	**Array**

Unit	Unit Outcomes (UOs) (4 to 6 UOs at different levels)	Topics and Sub-topics
Array and Pointers	one dimensional arrays in 'C' language. 4b. Write programs using two dimensional arrays 4c. Write programs to demonstrate the use of pointers in C programs. 4d. Write simple programs to demonstrate use of strings	4.1 Introduction to an Array 4.2 A characteristics of an array 4.3 One dimensional array: Declaration, initialization and accessing 4.4 Two-dimensional array: Declaration, initialization and accessing 4.5 Introduction to a String: Declaration and Initialization of String, gets() and puts() **Pointer** 4.6 Introduction to Pointers 4.7 Characteristics of Pointers 4.8 Address of Operator and Indirection operator 4.9 Declaration and initialization of Pointers 4.10 Types of Pointers: void and null 4.11 Pointers to Pointers
Unit–V: **Functions**	5a. Write a simple c program to declare, define and call a function. 5b. Write C programs using function with arguments 5c. Write C functions using call by value and call by reference. 5d. Write C programs using recursive functions. 5e. Use built-in functions of math and string library	5.1 Introduction to Functions 5.2 Types of Functions: Built-in and user defined Functions 5.4 Advantages of using Functions 5.5 Working of a Function 5.6 Declaring, Defining and calling user defined Functions 5.7 Categories of user-defined Functions 5.8 Call by Value and call by Reference 5.10 Recursion 5.11 Built-in Functions: String and Maths 5.12 Storage Classes: -auto, static, register and extern
Unit– VI: **Structure, Union and Files**	6a. Write a simple C program to define, declare and access user defined Structure 6b. Write a simple C program to define, declare and access user defined union 6c. Develop a program to read from and write into files using 'C' language 6d. Write a simple program to demonstrate use of "Array of structures"	**Structure** 6.1 User-defined Data types: enum, typedef 6.2 Introduction to Structures 6.3 Declaration, Initialization and accessing of Structures 6.4 Array of structures **Union** 6.5 Introduction to Union 6.6 Declaration, Initialization and accessing of Union **Files** 6.7 Introduction to text Files 6.8 Opening & Closing Files in text mode

Unit	Unit Outcomes (UOs) (4 to 6 UOs at different levels)	Topics and Sub-topics
		6.9 Reading From and writing into Files in text mode only

***Note**: The Unit Outcomes (UOs) need to be formulated at the 'Application Level' and above of Revised Bloom's Taxonomy' to accelerate the attainment of the COs and the competency.*

9. SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Flowchart and Algorithm	4	2	2	4	8
II	Basics of 'C'	6	2	6	2	10
III	Decision Statements and Loop Control Statements	10	2	6	10	18
IV	Array and Pointers	8	2	5	4	11
V	Functions	8	2	5	6	13
VI	Structure, Union and Files	6	2	4	4	10
Total		**42**	**12**	**28**	**30**	**70**

***Legends:** R=Remember, U=Understand, A=Apply and above (Revised Bloom's taxonomy)*

***Note**: This specification table provides general guidelines to assist students for their learning and to teachers to teach and question paper designers/setters to formulate test items/questions to assess the attainment of the UOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may slightly vary from above table.*

10. SUGGESTED STUDENT ACTIVITIES

Other than the classroom and laboratory learning, following are the suggested student-related ***co-curricular*** activities which can be undertaken to accelerate the attainment of the various outcomes in this course: Students should perform following activities in group and prepare reports of about 5 pages for each activity. They should also collect/record physical evidences for their (student's) portfolio which may be useful for their placement interviews:

a) Design algorithm and construct a flowchart for at least 6 problems
b) Students are encouraged to learn Visual Language programming like scratch, snap etc.
c) Undertake micro-projects in teams.
d) Prepare charts to explain use/process of the identified topic.
e) https://www.codechef.com/ , in this website very elementary programs are available, students are expected to solve those programs
f) https://code.org/, an hour of coding event may be organized and students are encouraged to participate.
g) Students are encouraged to register themselves in various MOOCs such as: Swayam, edx, Coursera, Udemy etc to further enhance their learning.
h) Encourage students to participate in different coding competitions like hackathon, online competitions on codechef etc.
i) Encourage students to form a coding club at institute level.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

These are sample strategies, which the teacher can use to accelerate the attainment of the various outcomes in this course:

a) Massive open online courses (***MOOCs***) may be used to teach various topics/sub topics.

b) Guide student(s) to take micro-projects.

c) Blend the basic concepts with more specialized instruction

d) Visualization, Cooperative Learning, inquiry based instruction, differentiation, effective use of technology, think-pair and share etc pedagogies can be implemented as per the enlisted course outcomes.

e) Give at least 10 competitive problems for each course outcomes of this course

f) Practice, practice and practice - expose students to wide range of problems

g) About ***20% of the topics/sub-topics*** which are relatively simpler or descriptive in nature is to be given to the students for ***self-learning***, but to be assessed using different assessment methods.

h) With respect to ***section No.10***, teachers need to ensure to create opportunities and provisions for ***co-curricular activities***.

i) Guide students on how to address issues on environment and sustainability using the knowledge of this course

12. SUGGESTED MICRO-PROJECTS

Only one micro-project is planned to be undertaken by a student that needs to be assigned to him/her in the beginning of the semester. In the first four semesters, the micro-project are group-based (group of 3 to 5). However, **in the fifth and sixth semesters**, the number of students in the group should ***not exceed three.***

The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. Each student will have to maintain dated work diary consisting of individual contribution in the project work and give a seminar presentation of it before submission. The duration of the microproject should be about **14-16** ***(fourteen to sixteen) student engagement hours*** during the course. The students ought to submit micro-project by the end of the semester to develop the industry-oriented COs.

A suggestive list of micro-projects is given here. This has to match the competency and the COs. Similar micro-projects could be added by the concerned course teacher:

Suggested List of Micro-Project Definition

1. Develop a C program to represent a bank account. Create a structure Customer having fields name of the depositor, account number, type of account and balance amount in the account. Perform different operations: (1) To assign initial values (2) To deposit an amount (3) To withdraw an amount after checking the balance (4) To display name and balance. Write a menu driven program to handle N number of customers.
2. Develop a menu driven C program to perform basic arithmetic operations/mathematical operations like calculators on user inputted data.
3. Develop a C program to generate results for students. Admin enters component wise marks for each subject. After entering the marks, students will know his/her SPI as well as total backlogs.

4. Develop a C program to display a minimum number of currency notes required based on the entered amount. Output will also display the total number of notes required for each currency note. Valid currency notes are 1, 2, 5, 10, 20, 50, 100, 200, 500, 2000. E.g. if the user enters 140 then the output will be "3 currency notes are required. 1*100 + 2*20 = 140".
5. Develop a C program that allows the names of 1000 candidates in a local election and the number of votes received by each candidate. The program should then output each candidate's name, the number of votes received, and the percentage of the total votes received by the candidate. Your program should also display the winner of the election.
6. Develop a C program to find and replace all occurrences of a word in file. For example : Suppose file contains : "I like programming. I am learning programming and programming with files is fun. Learning programming is simple and easy." Find occurrences of "programming" and replace it with "C language".

13. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication with place, year and ISBN
1	Programming with ANSI and Turbo C	Ashok N. Kamthane	Pearson Education, New Delhi; 2008; ISBN: 978-8131704370
2	Programming in ANSI C	E. Balagurusamy	McGraw Hills Education, New Delhi; 2019; ISBN: 978-9351343202
3	Let us 'C'	Yashavant Kanetkar	BPB Publication, New Delhi; 2020; ISBN: 978-9389845686
4	Introduction to C Programming	Reema Thareja	Oxford University Press, New Delhi; 2018; ISBN: 978-0199492282

14. SUGGESTED LEARNING WEBSITES

a) https://snap.berkeley.edu/snap/snap.html
b) https://scratch.mit.edu/download/scratch2
c) http://nptel.ac.in/courses/! 06105085/4
d) www.w3schools com
e) wvvw. program iz. com/c-programming
f) https://www codecademy.com/courses/getting-started-v2/0/l
g) http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-087-practical-programming-in-c-january-iap-2010/

http://spoken-tutorial org/

15. PO-COMPETENCY-CO MAPPING

Semester I	Computer Programming (Course Code: 4310702)						
	POs						
Competency & Course Outcomes	**PO 1 Basic & Discipline specific knowledge**	**PO 2 Problem Analysis**	**PO 3 Design/ develop-ment of solutions**	**PO 4 Engineering Tools, Experimen-tation &Testing**	**PO 5 Engineering practices for society, sustainability & environment**	**PO 6 Project Manage-ment**	**PO 7 Life-long learning**
Competency ***Develop structured, modular and memory efficient programs in 'C'.***							
Course Outcomes CO a) Design flowchart and algorithm for given programming statement.	2	2	2	-	2	2	2
CO b) Develop, Debug basic C programs, different operators, decision making controls and iterative statements.	2	3	3	2	2	2	3
CO c) Develop C programs using one dimensional arrays, dimensional arrays and pointers.	2	3	3	2	2	2	3
CO d) Implement types of user defined functions.	2	3	3	2	2	2	3
CO e) Exhibit use of structure and union in c language.	2	3	3	2	2	2	3
CO f) Implement file and I/O operations in C language.	2	3	3	2	2	2	3

Legend: '**3'** for high, '**2**' for medium, **'1'** for low and **'-'** for no correlation of each CO with PO.

16. COURSE CURRICULUM DEVELOPMENT COMMITTEE

GTU Resource Persons

Sr. No.	Name and Designation	Institute	Contact No.	Email
1.	Shri P. P. Kotak Principal	Government Polytechnic, Rajkot	9825469617	kotakp2003@yahoo.com
2.	Shri K. N. Raval HOD	DTE, Gandhinagar	9427952082	raval.kamlesh@gmail.com
3.	Smt. H. B. Kotak HOD	A.V. Parekh Technical Institute, Rajkot	9429048096	kotakhemali@gmail.com
4.	Smt. M. P. Mehta HOD	Government Polytechnic, Gandhinagar	9879578273	manishamehtain@gmail.com
5.	Smt. M. V. Prajapati Sr. Lecturer	Government Polytechnic, Gandhinagar	9428049861	mvprajapati2014@gmail.com
6.	Smt. J. J. Karagthala Lecturer	Government Polytechnic For Girls, Ahmedabad	9824799620	jdaftary@gmail.com
7.	Shri K. M. Madhu Lecturer	K. D. Polytechnic, Patan	9662402343	Kaushalmadhu.cse@gmail.com

NITTTR Resource Persons

Sr. No.	Name and Designation	Department	Contact No.	Email
1.	Dr. R. K. Kapoor	Department of Commuter Science and Engineering Education	0755-2661600 (Ext393)	rkkapor@nitttrbpl.ac.in
2.	Dr. Sanjay Agrawal	Department of Commuter Science and Engineering Education	0755-2661600 (Ext392)	sagrawal@nitttrbpl.ac.in