GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)

Competency-focused Outcome-based Green Curriculum-2021 (COGC-2021)
Semester-II

Course Title: Programming Methodology
(Course Code: 4321703)

Diploma programme in which this course is offered	Semester in which offered
Instrumentation and Control Engineering	Second

1. RATIONALE

The course deals with the analysis of problems, design of solution and implementation as algorithm. Rather than training the students on syntax of any particular programming language, a study of programming should focus on problem-solving - in a rigorous, systematic fashion - through algorithms. Thus, the primary focus is on the design of algorithms for solutions to logical problems.

2. COMPETENCY

Study of programming should enable students to develop their problem solving approach. Student should be able to develop and test logic using common constructs of programming languages, rather than start study with any programming language first. A study in this perspective offers the flexibility of choosing any language to code the designed algorithm.

3. COURSE OUTCOMES (COs)

On completion of this course, the student should be able to:

a) Identify various common paradigms of problem solving tools/techniques.
b) Develop flow chart and algorithm for the solution to logical problems.
c) Use common constructs shared by programming languages.
d) Test logic solution for real life problems using any execution tool.
e) Identify steps for optimized use of iteration considering environmental benefits.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme (In Hours)			Total Credits (L+T+P/2)	Examination Scheme				
				Theory Marks		Practical Marks		Total Marks
L	T	P	C	CA	ESE	CA	ESE	
0	0	4	2	00	00	25*	25	50

(*): For this practical only course, 25 marks under the practical CA have two components i.e. the assessment of micro-project, which will be done out of 10 marks and the remaining 15 marks are for the assessment of practical. This is designed to facilitate attainment of COs holistically, as there is no theory ESE.

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

5. SUGGESTED PRACTICAL EXERCISES

Following practical outcomes (PrOs) are the sub-components of the Course Outcomes (Cos). Some of the **PrOs** marked **'*'** are compulsory, as they are crucial for that particular CO at the 'Precision Level' of Dave's Taxonomy related to 'Psychomotor Domain'. *Practical number 3 to 7, 11 to 23 and 25 can be performed using any logic execution tools/programming language.*

Sr. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
1	*Study different problem solving techniques and tools.	1	2
2	*Study different data types and variables used in problem solving tools e.g. Integer, Float, Character, Boolean, String	1	2
3	*Develop logic with for minimum two problems to understand Data types.	1	2
4	*Develop logic for minimum two problems using Constants and Variables	1	2
5	*Develop logic for minimum two problems to understand Arithmetic operators.	1	2
6	*Develop logic for minimum two problems providing understanding of Relational operators	1	2
7	*Develop logic for minimum two problems using logical and bitwise operators	1	2
8	*Study flowchart and its symbols	1	2
9	*Draw Flow Chart for at least four problems.	2	4
10	*Develop algorithm for at least four problems.	2	4
11	Develop logic with flowchart for minimum two problems using simple If statements	3	2
12	*Develop logic with flowchart for minimum two problems using if else statements.	3	2
13	*Develop logic with flowchart for minimum two problems using If-else-if statements	3	2
14	Develop logic with flowchart for minimum two problems using a Nested If statements	3	2
15	*Develop logic with flowchart for minimum two problems using While Loop.	3	2
16	Develop logic with flowchart for minimum two problems using Do-while loop.	3	2
17	*Develop logic with flowchart for minimum two problems to understand simple For loop and nested For loop.	3	2
18	*Develop logic with flowchart for minimum two problems to understand break, continue, statements.	3	2
19	Develop logic with flowchart for minimum two problems to understand goto and switch statements.	3	2
20	*Develop logic with flowchart for minimum two problems involving one dimensional array.	4	2
21	Develop logic with flowchart for minimum two problems involving two dimensional arrays.	4	2

Sr. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
22	*Develop logic with flowchart for minimum two problems using Function –Call by value and reference.	4	2
23	*Develop logic with flowchart for minimum two problems using recursion.	4	2
24	*To study level of optimization and techniques	5	2
25	Develop and compare 2 algorithm of a logical problem using iteration structure for efficient use of optimization.	5	4
	Minimum 24 Practical Exercises		**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. *Care must be taken in assigning and assessing study report as it is a first year study report. Study report, data collection and analysis report must be assigned in a group. Teacher has to discuss about type of data (which and why) before group start their market survey.*

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.

Sr. No.	Sample Performance Indicators for the PrOs	Weightage in %
1	Correctness of algorithm/flowchart	20
2	Readability and documentation of the algorithm/flowchart of input and output displayed (messaging and formatting)	20
3	Code efficiency	10
4	Debugging ability	20
5	Logic execution/answer to sample questions	20
6	Optimized use of iteration	10
Total		**100**

6. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

These major equipment with broad specifications for the PrOsis a guide to procure them by the administrators to user in uniformity of practical's in all institutions across the state.

Sr. No.	Equipment Name with Broad Specifications	PrO. No.
1.	Computer with basic configuration with windows or Unix OS	All
2.	Logic execution tools/software like C Compiler, Scilab, Arduino IDE, Python, Matlab, tinkercad etc.	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 fulfill the development of this course competency.

a) Work as a leader/a team member.
b) Follow problem solving approach in real life.
c) Realize importance of Energy Conservation. (Environment related)

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 **Introduction to Problem Solving & Programming**	1a. List the steps in the general problem solving strategy. 1b. Discuss different problem solving techniques and tools 1c. Explain basic components of program- input, processing data, output 1d. Identify and select from different data types and specific operations.	**1.1 Introduction to programming** General Problem solving, Steps of problem solving, Classification of computer languages, Language translators (Assembler, Compiler, Interpreter), Linker, Characteristics of a good programming language, Factors for selecting a language **1.2 Basic programming concept**: Input, Processing Data, Output Data; **1.3 Data types, Variables and constants :** Integer, Float, Character, Boolean, String etc **1.4 Type specific operations:** arithmetic, relational, logical operations, Built-in functions for operation.

Unit	Unit Outcomes (UOs) (4 to 6 UOs at different levels)	Topics and Sub-topics
Unit– II **Flowchart and Algorithm**	2a. Draw flow chart to solve given program logically 2b. Apply heuristic solutions to problems through illustration of examples 2c. Apply algorithmic solutions to problems through illustration of examples, discussion	2.1 **Flow chart:** ● Definition and Importance of flowchart. ● Symbols of Flowchart. ● Flow lines, Terminals, Input/Output ,Processing Decision, Connection off-page connectors ● Guidelines for preparing Flowchart. ● Flowchart structure: Sequence, selection, repetition. ● Limitation of flowchart Emergency action Plan 2.2 **Algorithm and heuristic** ● Developing and writing algorithm ● Heuristic approach
Unit-III **Control Structures**	3a. Explain multiple alternative structures using flowchart and algorithm. 3b. Apply IF structure to construct program. 3c. Apply Case structure to construct program. 3d. Apply Repetition (iteration) Structures to solve the problem.	3.1 **Introduction to Selection Structures**: ● Single ,dual, multiple alternative structures ● IF structure: Simple IF, IF-Else, Nested IF & IF-ladder ● Case structure 3.2 **Introduction to Repetition(iteration) Structures:** ● Basics of Loop, Types of loops, Pre-test and post-test loops, While loop, Repeat-until loop, For loop ● Applications of repetition structures, Problem solving with repetition structures, examples
Unit-IV **Array , Pointer and Function**	4a. Explain the need and use of arrays 4b. Use one dimensional arrays through illustration, problem solving	**4.1 Introduction of Array and Array initialization**: ● Characteristics of an array, One dimensional array, Operations to one dimensional array, Two

Unit	Unit Outcomes (UOs) (4 to 6 UOs at different levels)	Topics and Sub-topics
	4c. Use two dimensional arrays through illustration, problem solving 4d. Use function for problem solving	dimensional arrays (multidimensional), Operations to two dimensional **4.2 Pointer & Function** ● Basic of pointer ● Basic of function, definition and return ● Types of functions-call by value and reference ● Recursion-Types of recursion, Rules for recursive function, direct and indirect recursion, Advantages and disadvantages of recursion
Unit-V **Program Optimization and Case studies**	5a. Define optimization; 5b. Classify types of optimization 5c. State levels of optimization. 5d. Analyze efficient use of algorithm for developing application through case studies.	**5.1 Introduction to optimization** ● Types of optimization ● Levels of optimization **5.2 Case studies of algorithm for developing application**

9. SUGGESTED SPECIFICATION TABLE FOR QUESTIONPAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Introduction to Problem Solving & Programming	**Not Applicable**				
II	Flowchart and Algorithm					
III	Control Structures					
IV	Array , Pointer and Function					
V	Program Optimization and Case studies					

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

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) Give seminar on reading a real life problem solving techniques.
c) Undertake a market survey of different industry suited programming languages.
d) Students are encouraged to learn Visual Language programming like scratch, snap etc
e) Students are encouraged to register themselves in various MOOCs such as: Swayam, edx, Coursera, Udemy etc to further enhance their learning
f) Prepare the Charts that classify recycling process for electronic waste and plastics.

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) in undertaking micro-projects.
C. 'L' in section No. 4means different types of teaching methods that are to be employed by teachers to develop the outcomes.
D. 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.
E. With respect to section No.10, teachers need to ensure to create opportunities and provisions for co-curricular activities.
F. Introduce E-waste recycling technology among the students.
G. Guide students for reading data sheets.

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-projects 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 micro project 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:

A. Prepare comparison report/presentation on different programming languages with advantages and disadvantages.
B. Prepare report/presentation on latest trends in programming languages well suited for industries.

C. Prepare report/presentation on different optimization tools and/or techniques in programming languages.
D. Prepare chart to explain different real life problem solving tools and/or techniques.
E. Prepare program for simple calculator.
F. Prepare program for hospital management system.
G. Prepare program for parking management system.
H. Prepare program for automatic traffic control system.

13. SUGGESTED LEARNING RESOURCES

Sr. No.	Title of Book	Author	Publication with place, year and ISBN
1	Problem Solving and Programming Concepts, 9th Ed.	Maureen Sprankle, Jim Hubbard	Pearson Publication ISBN 978-0-13-249264
2	Programming in ANSI C	E. Balagurusamy	McGraw Hills Education, New Delhi; 2019; ISBN: 978-9351343202
3	Computer science with python	Sumita Arrora	Dhanpat rai & sons
4	Introduction to Scilab	Sandeep Nagar	Apress ISBN: 9781484231920
5	Practical Programming: An Introduction to Computer Science Using Python 3.6, THIRD EDITION	Paul Gries, Jennifer Campbell, Jason Montojo	The Pragmatic Bookshelf

14. SOFTWARE/LEARNING WEBSITES

- Tinkercad – 3D design, electronics, and coding open source software
- MIT APP Inventor
- Scilab open Source software
- Programming in C
- Arduino IDE
- https://en.wikipedia.org/wiki/Program_optimization

15. PO-COMPETENCY-CO MAPPING

Semester II	Programming Methodology (Course Code: 4321703)						
	POs						
Competency & Course Outcomes	**PO 1 Basic & Discipline specific knowledge**	**PO 2 Problem Analysis**	**PO 3 Design/ development of solutions**	**PO 4 Engineering Tools, Experimentation &Testing**	**PO 5 Engineering practices for society, sustainability & environment**	**PO 6 Project Management**	**PO 7 Life-long learning**
Competency	Study of programming should enable students to develop their problem solving approach. Student should be able to develop and test logic using common constructs of programming languages, rather than start study with any programming language first. A study in this perspective offers the flexibility of choosing any language to code the designed algorithm..						
Course Outcomes CO 1) Identify various common paradigms of problem solving tools/techniques.	3	1	-	1	1	-	1
CO 2) Develop flow chart and algorithm for the solution to logical problems.	2	2	2	1	1	1	1
CO 3) Use common constructs shared by programming languages.	2	1	2	1	1	1	1
CO 4) Test logic solution for real life problems using any execution tool.	2	3	3	3	1	1	1
CO 5) Identify steps for optimized use of iteration considering environmental benefits.	2	1	1	1	3	1	2

*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

S. No.	Name and Designation	Institute	Contact No.	Email
1.	Shri Manan A. Modi, Lecturer IC- Engineering	Government Polytechnic, Ahmedabad	9974244507	manan1272000@gmail.com
2.	Shri Jiten B. Maheta Lecturer IC- Engineering	Government Polytechnic, Gandhinagar	9979901567	jiten.maheta@gmail.com
3.	Ripal M. Pathak Lecturer IC- Engineering	AVPTI, Rajkot	9428862255	ripalpathak@gmail.com