

# **INSTITUTE OF AERONAUTICAL ENGINEERING**

(Autonomous) Dundigal, Hyderabad - 500 043

# COMPUTER SCIENCE AND ENGINEERING

# **COURSE DESCRIPTION FORM**

Course Title	Computer Programming									
Course Code	ACS001									
Regulation	IARE-R16									
C St t	Lectures	Practicals	Credits							
Course Structure	3	-	-	3						
Course Coordinator	Dr. K. Srinivasa Reddy, Professor									
Team of Instructors	Dr. G. Ramu, Professor, Ms. B. Pac Ms. P. Ila Chandana Kumari , Asso Ms. K. Laxmi Narayanamma , Asso	lmaja, Associate Pr ciate Professor, ciate Professor, Ms	ofessor . B Rekha , Assistar	nt Professor						

#### I. COURSE OVERVIEW

The course covers the basics of programming and demonstrates fundamental programming techniques, customs and terms including the most common library functions and the usage of the preprocessor. This course helps the students in gaining the knowledge to write simple C language applications, mathematical and engineering problems. This course helps to undertake future courses that assume this programming language as a background in computer programming. Topics include variables, data types, functions, control structures, pointers, strings, arrays and dynamic allocation principles. This course in reached to student by power point presentations, lecture notes, and lab involve the problem solving in mathematical and engineering areas.

#### II. **PREREQUISITE(S)**

Level	Credits	Periods/ Week	Prerequisites
UG	3	3	Fundamentals of Computers, Intelligence to do complex problem-solving

### III. MARKS DISTRIBUTION

Subject	SEE Examination	CIA Examination	Total Marks
Computer Programming	70 Marks	30 Marks	100 Marks

Semester End Examination70 Marks5 qu70 Marks(3 Hours)	o questions to be answered. Each question carries 14 Marks
---	---

	Continuous Internal Assessment - 1							
				Part - A				
			Continuous Internal	5 questions to be answered out of				
			Examination (CIE)	5 questions, each carries 1 mark.				
	30 Marks	Units	(2 hours)	Part - B				
	(2 Hours)	I, II and	(2 110013)	4 questions to be answered out of				
		III (half)		5 questions, each carries 5 marks.				
			Quiz-I /Alternate					
			Assessment Tool	5 marks for assignment.				
A			(AAT-I)	_				
CIA Examinations		Co	ssment - 2					
CIA Examinations				Part – A				
			Continuous Internal	5 questions to be answered out of				
			Examination (CIE)	5 questions, each carries 1 mark.				
	20 Mortes	Units	(2 hours)	Part - B				
	(2  Hours)	III (half)	(2 110013)	4 questions to be answered out of				
	(2 Hours)	IV and V		5 questions, each carries 5 marks.				
			Ouiz-II /Alternate					
			Assessment Tool	5 marks for assignment.				

#### IV. **EVALUATION SCHEME**

S. No	Component	Duration	Marks				
1	CIE - I Examination	2 hour	25				
2	Quiz - I / AAT - I	-	05				
	TOTAL	30					
3	CIE - II Examination	2 hour	25				
4	Quiz - II / AAT - II	-	05				
	TOTAL		30				
	CIA Examination marks to be considered as average of above two CIA's						
5	EXTERNAL Examination	3 hours	70				
	GRAND TOTAL	100					

#### V. **COURSE OBJECTIVES**

- i. Learn adequate knowledge by problem solving techniques.
- ii. Understand programming skills using the fundamentals and basics of C Language.iii. Improve problem solving skills using arrays, strings, and functions.
- iv. Understand the dynamics of memory by pointers.
- v. Study files creation process with access permissions.

#### VI. COURSE OUTCOMES

#### At the end of the course the students are able to:

- 1. **Identify** and understand the working of key components of a computer system.
- 2. Develop algorithms and flowcharts for solving mathematical and engineering problems.
- 3. Design programs involving decision structures, loops and functions.
- 4. Use structured data types and the concept of arrays in simple data processing applications.
- Differentiate between static and dynamic memory allocation.
- 6. **Understand** the concept of recursion and describe its implementation using a stack.
- 7. Understand pointers, memory allocation and data handling through files in C.
- 8. Develop and execute solutions to solve real-time applications using the features of C language.
- 9. **Build** confidence for self learning needed for computer languages.

### VII. HOW PROGRAM OUTCOMES ARE ASSESSED

	Program Outcomes	Level	Proficiency assessed by
PO1	<b>Engineering knowledge</b> : Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	Н	Assignments, Tutorials
PO2	<b>Problem analysis</b> : Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	Н	Assignments
PO3	<b>Design/development of solutions</b> : Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	S	Mini Projects
PO4	<b>Conduct investigations of complex problems</b> : Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	S	Projects
PO5	<b>Modern tool usage</b> : Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	S	Mini Projects
PO6	<b>The engineer and society</b> : Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	N	-
PO7	<b>Environment and sustainability</b> : Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	Ν	
PO8	<b>Ethics</b> : Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	Ν	
PO9	<b>Individual and team work</b> : Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	N	
PO10	<b>Communication</b> : Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	N	
PO11	<b>Project management and finance</b> : Demonstrate knowledge and understanding of the engineering and management principles and apply	Ν	

	Progr	Level	Proficiency assessed by		
	these to one's own work, as projects and in multidisciplin				
PO12	<b>Life-long learning</b> : Recogn ability to engage in indepe context of technological char	S	Projects		
-	N - None	S - Supportive	H - Highly Related	1	_

# VIII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED

	Program Specific Outcomes	Level	Proficiency assessed by
PSO1	<b>Professional Skills:</b> The ability to research, understand and implement computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer-based systems of varying complexity.	Н	Lectures, Assignments
PSO2	<b>Problem-solving Skills:</b> The ability to apply standard practices and strategies in software project development using open-ended programming environments to deliver a quality product for business success.	S	Projects
PSO3	<b>Successful Career and Entrepreneurship:</b> The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.	S	Guest Lectures

N - None S - Supportive H- Highly Related

#### IX. SYLLABUS

# UNIT – I

# INTRODUCTION

Introduction to computers: Computer systems, computing environments, computer languages, creating and running programs, algorithms, flowcharts; Introduction to C language: History of C, basic structure of C programs, process of compiling and running a C program, C tokens, keywords, identifiers, constants, strings, special symbols, variables, data types; Operators and expressions: Operators, arithmetic, relational and logical, assignment operators, increment and decrement operators, bitwise and conditional operators, special operators, operator precedence and associativity, evaluation of expressions, type conversions in expressions, formatted input and output.

#### UNIT – II

#### CONTROL STRUCTURES, ARRAYS AND STRINGS

Control structures: Decision statements; if and switch statement; Loop control statements: while, for and do while loops, jump statements, break, continue, goto statements; Arrays: Concepts, one dimensional arrays, declaration and initialization of one dimensional arrays, two dimensional arrays, initialization and accessing, multi dimensional arrays; Strings concepts: String handling functions, array of strings.

#### UNIT – III

#### FUNCTIONS AND POINTERS

Functions: Need for user defined functions, function declaration, function prototype, category of functions, inter function communication, function calls, parameter passing mechanisms, recursion, passing arrays to functions, passing strings to functions, storage classes, preprocessor directives.

Pointers: Pointer basics, pointer arithmetic, pointers to pointers, generic pointers, array of pointers, pointers and arrays, pointers as functions arguments, functions returning pointers.

#### UNIT – IV

#### STRUCTURES AND UNIONS

Structures and unions: Structure definition, initialization, accessing structures, nested structures, arrays of structures, structures and functions, passing structures through pointers, self referential structures, unions, bit fields, typedef, enumerations; Dynamic memory allocation: Basic concepts, library functions.

### UNIT – V

#### FILES

Files: Streams, basic file operations, file types, file opening modes, file input and output functions, file status functions, file positioning functions, command line arguments.

#### TEXT BOOKS

- 1. Stephen G. Kochan, "Programming in C", Addison-Wesley Professional, 4th Edition, 2014.
- 2. B.A.Forouzan and R.F. Gilberg, "Computer Science: A Structured Programming Approach Using C", 3e, Cengage Learning.

#### **REFERENCE BOOKS**

- W. Kernighan Brian, Dennis M. Ritchie, "The C Programming Language", PHI Learning, 2<sup>nd</sup> Edition, 1988.
- 2. Yashavant Kanetkar, "Exploring C", BPB Publishers, 2<sup>nd</sup> Edition, 2003.
- 3. E. Balagurusamy, "Programming in ANSI C", Mc Graw Hill Education, 6<sup>th</sup> Edition, 2012.
- 4. Schildt Herbert, "C: The Complete Reference", Tata Mc Graw Hill Education, 4<sup>th</sup> Edition, 2014.
- 5. R. S. Bichkar, "Programming with C", Universities Press, 2<sup>nd</sup> Edition, 2012.
- 6. Dey Pradeep, Manas Ghosh, "Computer Fundamentals and Programming in C", Oxford University Press, 2<sup>nd</sup> Edition, 2006.

## X. COURSE PLAN

At the end of the course, the students are able to achieve the following course learning outcomes.

	Lecture	Topics to be covered	Course Learning Outcomes	Reference
	1-2	Introduction to Computers: computer	Identify basic parts of	T2:1.1-1.2
I	3 – 4	Computer languages, creating and running programs, program development.	<b>Develop</b> algorithms and flowcharts for given problem.	T1:2
	5-6	<b>Introduction to C Language</b> : background, C programs.	<b>Understand</b> the basics of a C programming.	T2:2.1-2.2
	7 – 8	Identifiers, data types, Input/output, variables, constants, Operators (arithmetic, relational, logical, bitwise etc).	<b>Understand</b> the basic building blocks of a C program.	T1:4
	9 - 10	Expressions, precedence and associativity, expression evaluation, type conversions.	<b>Understand</b> the rules of precedence and associativity in evaluating expressions.	T1:4
	11 – 12	Statements - selection statements (making decisions) – if statement, switch statement.	Write programs using two- way and multi-way selection.	T1: 6
П	13 – 14	Repetition statement (loops)-while, do- while statements, for statements, loop examples.	<b>Identify</b> appropriate loop	T1: 5
	15 – 16	Other statements related to looping – break, continue, go to, simple C program examples.	problem.	T1:5 T1:17
	17 – 18	<b>Arrays-</b> Concepts, using arrays in C, declaration and initialization of one dimensional array, C program example. Two dimensional arrays, initialization and accessing, multi dimensional arrays, C program example.	<b>Understand</b> the usage of arrays to store homogenous data. Understand the applications of arrays	T1:7
	19 – 20	Strings – Strings concepts: String handling functions, array of strings, C program examples.	Write programs to manipulate strings.	T1:10
	21 – 24	<b>Functions</b> - Need for user defined functions, function declaration, function prototype. Category of functions, inter function communication, function calls, parameter passing mechanisms.	<b>Design</b> and <b>implement</b> multi-function programs.	T1:8
ш	23 - 25	Recursion, passing arrays to functions, passing strings to functions, Storage classes and preprocessor commands.	<b>Implement</b> recursive solutions to problems and passing arrays and strings as parameters and differentiate local and global scope of variables and write programs using preprocessor directives.	T1:8 T1:13
	26 27	Pointer basics, pointer arithmetic, pointers to pointers, generic pointers. Pointer applications-Arrays and pointers, pointer arithmetic and arrays, passing an array to a function.	<b>Understand</b> the basic concepts of pointers. Write effective programs using pointers.	T1:8

		28 – 29	Array of pointers, pointers and arrays, pointers as functions arguments, functions returning pointers	Write effective programs using pointers.	T1:11
		30 - 31	Structures – declaration, initialization, accessing structures, operations on structures.	<b>Understand</b> the usage and	T1:9
		32 - 33	Complex structures, structures and functions, passing structures through pointers, self-referential structures.	store heterogeneous data.	T1: 9
I	V	34 – 35	Unions, C programming examples, Bit fields, typedef, enumerations.	<b>Differentiate</b> structures and unions in terms of memory allocation and understand, bit fields, the enumerated data types.	T1:9 T1:14 T1: 17
	-	36 38	Dynamic memory allocation: Basic concepts, library functions.	Write effective programs using pointers for dynamic memory allocation.	T1: 17
	-	39 – 40	<b>Files:</b> Concept of a file, streams, types of files and file opening modes.	<b>Understand</b> the basic properties and characteristics of files.	T1:16
		41-42	File input/output functions (standard input/output functions for files).	<b>Use</b> library file I/O functions for storing data on secondary storage.	T1:16
	V	43 – 44	File status functions (error handling), positioning functions, C program examples.	Write programs to handle simple file I/O errors.	T1: 16
		45	Command-line arguments.	<b>Understand</b> the use of command-line arguments.	T1:17

### XI MAPPING COURSE OBJECTIVES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES

Course Objectives	Program Outcomes								Program Specific Outcomes						
Objectives	POI	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	<b>PO10</b>	PO11	PO12	PSO1	PSO2	PSO3
I	Н	Н										S	Н	Н	
II		Η	S		S								S	Н	
1111	S	Η	S										Н		S
IV	Н	S											Н	S	
V	Н	S											Н	S	S
S -	S - Supportive H - Highly Related														

### XII MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES ANDPROGRAM SPECIFIC OUTCOMES

Course Outcomes	Program Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	<b>PO10</b>	PO11	PO12	PSO1	PSO2	PSO3	
1	Н			S									Н	S		
2	S	Η	S									S	Н		S	
3	S	Η			S								S			
4	Н			S								S	S	S		
5	S	Н	S		S							S	Н		S	
6	Н	S		S									S			
7	Н				S							S	Н	S		
8	Н			S									Н	S		
9	S	Н	S									S	Н		S	
S - Supportive						H - Highly Related										

Prepared by: Ms. B Padmaja, Associate Professor

#### HOD, COMPUTER SCIENCE AND ENGINEERING