Department of Freshman Engineering

Programming for Problem Solving

Course Category Engineering Science Branch IT Course Type Theory Credits 3 L-T-P 3-0-0 Prerequisites Problem Solving Techniques Continuous 30 Semester End Evaluation 70 Total Marks 100 Internal Evaluation Evaluation Marks 100 Upon successful completion of the course, the student will be able to COI Understand the principles of structured programming and C constructs for solving problems. (L2) CO2 Apply suitable control constructs and array concepts to solve problems. (L3) CO4 CO4 Analyze the given problem and use modular programming approach to develop solutions. (L4) Contribution of Course Outcomes & Strength of correlations (3:High, 2: Medium, 1:Low) PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO11 PO12 PS01 PS02 CO3 3 Image Image Syllabus Mapped CO's PS02 CO1 3 Image Image Image Image Col C0's Syllabus Mapped CO's	Course		20ES1202		Yea	Year		Ι		Sem	Semester		II			
Category Šcience Science Freequisites Problem Credits 3 L-T-P 3-0-0 Prerequisites Problem Continuous 30 Semester End 70 Total 100 Internal Evaluation Marks 100 Evaluation Course Outcomes 100 Q01 Understand the principles of structured programming and C constructs for solving problems. (L2) CO2 Q02 Apply the concept of pointers, user defined data types and files to solve problems. (L3) CO4 CO1 Understand the principles of structured programming approach to develop solutions. (L4) Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (SHigh, 2: Medium, 1:Low) PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO1 PO1 PS2 2 2 CO3 Apply the concept of pointers, User defined tata types and files to solutomes & Strength of correlations (SHigh, 2: Low) Tentrolutomes & Strength of correlations (SHigh, 2: Low) Tentrolutomes & Strength of correlations (SHigh, 2: Low) 2 2 2 2 2 2 2 2 2 2 2	Code				Deres			TT		C						
Credits 3 L-T-P 3-0-0 Prerequisites Problem Solving Techniques Continuous 30 Semester End Evaluation 70 Total Marks 100 Evaluation Course Outcomes 100 100 Upon successful completion of the course, the student will be able to 0 0 CO1 Understand the principles of structured programming and C constructs for solving problems. (L3) 0 CO2 Apply suitable control constructs and array concepts to solve problems. (L3) 0 0 CO3 Apply the concept of pointers, user defined data types and files to solve problems. (L4) 0 0 CO4 Analyze the given problem and use modular programming approach to develop solutions. (L4) 0 0 CO1 PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO1 PO1 PO1 PO1 PO1 PO1 1 1 CO2 3 Image Syllabus Mapped CO's 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 <td colspan="2"></td> <td colspan="2"></td> <td>Brai</td> <td colspan="2">Branch</td> <td colspan="2"></td> <td>Cou</td> <td colspan="2">Course Type</td> <td colspan="2">Theory</td>					Brai	Branch				Cou	Course Type		Theory			
Continuous 30 Semester End Fvaluation 70 Total Marks Solving Techniques Upon successful completion of the course, the student will be able to 100 100 CO1 Understand the principles of structured programming and C constructs for solving problems. (L2) 202 Apply suitable control constructs and array concepts to solve problems. (L3) 30 CO3 Apply the concept of pointers, user defined data types and files to solve problems. (L3) CO4 Analyze the given problem and use modular programming approach to develop solutions. (L4) Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2: Medium, 1:Low) PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO1 PO1 PS02 2 2 CO3 3 0 0 0 1 1 1 1 CO2 3 0 0 PO1 PO2 PS01 PS02 2 2 CO4 3 0 0 1 1 1 1 1 CO2 3 0 0 1 1 1 1 CO2 2 2 2 2 2 2 2 CO4					L.T.	L-T-P		3-0-0		Prer	Prerequisites		Problem			
Continuous Internal Evaluation 30 Semester End Evaluation Total Marks Total I00 Evaluation Course Outcomes 100 Evaluation Course Outcomes 100 Upon successful completion of the course, the student will be able to CO Understand the principles of structured programming and C constructs for solving problems. (L2) C01 Understand the principles of structured programming approach to develop solutions. (L4) CO C03 Apply the concept of pointers, user defined data types and files to solve problems. (L3) CO C04 Analyze the given problem and use modular types and files to solve problems. (L3) CO C04 PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO1 PO1 PO2 PO3 PO4 PO1 PO1 PO2 PO3 PO1 PO1 PO2 PO3 PO1 PO1 PO2 PO3 PO1 PO2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 <			5			1/-1-1		3-0-0								
Continuous 30 Semester End Evaluation 70 Total Marks 100 Evaluation Course Outcomes Impose Impose </td <td colspan="2"></td> <td colspan="2"></td> <td></td> <td colspan="2"></td> <td colspan="2"></td> <td></td> <td colspan="2"></td> <td colspan="2">-</td>													-			
Evaluation Course Outcomes Upon successful completion of the course, the student will be able to CO1 Understand the principles of structured programming and C constructs for solving problems. (L2) CO2 Apply suitable control constructs and array concepts to solve problems. (L3) CO3 Apply the concept of pointers, user defined data types and files to solve problems. (L3) CO4 Analyze the given problem and use modular programming approach to develop solutions. (L4) CO1 PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO11 PO12 PS01 PS02 CO3 3 Image: Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2: Medium, 1:Low) PO1 <po2 po10="" po11="" po12="" po3="" po4="" po5="" po6="" po7="" po8="" po9="" ps01="" ps02<="" td=""> 2 CO3 3 Image: Contribution of Course Syllabus Image: Contribution of Course Syllabus Unit No. Syllabus Mapped CO's Image: Contribution of C: Introduction, Structure of C Program, A Simple C Program, C-Tokens, Basic Data types, Variables, Constants, Input / Output statements, Operators, Type conversion and Type casting. CO1, CO2 2 Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements. Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. CO1, CO2</po2>	Conti	inuoi	15	30		Semester End		70		Tota	Total					
Course Outcomes Upon successful completion of the course, the student will be able to CO1 Understand the principles of structured programming and C constructs for solving problems. (L2) CO2 Apply suitable control constructs and array concepts to solve problems. (L3) CO3 Apply the concept of pointers, user defined data types and files to solve problems. (L3) CO4 Analyze the given problem and use modular programming approach to develop solutions. (L4) Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2: Medium, 1:Low) PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO12 PS01 PS02 CO1 3 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1	Inter	nal				Eval					Mar	Marks				
Upon successful completion of the course, the student will be able to CO1 Understand the principles of structured programming and C constructs for solving problems. (L2) CO2 Apply suitable control constructs and array concepts to solve problems. (L3) CO3 Apply the concept of pointers, user defined data types and files to solve problems. (L3) CO4 Analyze the given problem and use modular programming approach to develop solutions. (L4) Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2: Medium, 1:Low) PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO11 PO12 PSO1 PSO2 CO1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2	Evalu	atio	n													
CO1 Understand the principles of structured programming and C constructs for solving problems. (L2) CO2 Apply suitable control constructs and array concepts to solve problems. (L3) CO3 Apply suitable control constructs and array concepts to solve problems. (L3) CO4 Analyze the given problem and use modular programming approach to develop solutions. (L4) Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2: Medium, 1:Low) PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO12 PS01 PS02 CO3 3																
CO2 Apply suitable control constructs and array concepts to solve problems. (L3) CO1 Control constructs and array concepts to solve problems. (L3) CO4 Analyze the given problem and use modular programming approach to develop solutions. (L4) Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2: Medium, 1:Low) PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO1 PO1 PO12 PS01 PS02 CO1 3 Co1 1 1 1 1 1 1 1 1 1 1 1 1 Mapped CO's Syllabus Mapped CO's 1 Nutroduction to C: Introduction, Structure of C Program, A Simple C Program, C-Tokens, Basic Data types, Variables, Constants, Input / Output statements, Operators, Type conversion and Type casting. CO1, CO2 Conditional Branching Statements: if, if-else, if-else-if Statements and Switch case. Co1, CO2 2 <	-			_												
CO3 Apply the concept of pointers, user defined data types and files to solve problems. (L3) Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2:Medium, 1:Low) PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO11 PO12 PS01 PS02 CO1 3 Image: Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2:Medium, 1:Low) PO1 PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO11 PO12 PS01 PS02 CO1 3 Image: Course of Course Outcomes Strength of correlations (3:High, 2:Medium, 1:Low) Image: Course Outcomes Strength of correlations (3:High, 2:Medium, 1:Low) Image: Course Outcomes Strength of correlations (3:High, 2:Medium, 1:Low) Image: Course Outcomes Strength of correlations (3:High, 2:Medium, 1:Low) Image: Course Outcomes Strength of correlations (3:High, 2:Medium, 1:Low) Image: Course Outcomes Strength of correlations (3:High, 2:Medium, 1:Low) Image: Course Outcomes Strength of correlation (3:High, 2:Medium, 1:Low) Image: Course Outcomes Strength of correlation (3:High, 2:Medium, 1:Low) Image: Course Outcomes Strength of correlation (3:High, 2:Medium, 1:Low) Image: Course Outcomes Strength of correlation (3:H										ns. (L2)						
CO4 Analyze the given problem and use modular programming approach to develop solutions. (L4) Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2: Medium, 1:Low) PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO11 PO12 PS01 PS02 CO1 3 1 1 1 CO2 3 2 2 2 CO3 3 3 2 2 2 CO3 3 3 3 2 2 2 CO4 3 3 3 2 2 2 CO4 3 3 3 2 2 2 CO3 1 Introduction to C: Introduction, Structure of C Program, A Simple C Program, C-Tokens, Basic Data types, Variables, Constants, Input / Output statements, Operators, Type conversion and Type casting. CO1, CO2																
Contribution of Course Outcomes towards achievement of Program Outcomes & Strength of correlations (3:High, 2: Medium, 1:Low) PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO11 PO12 PS01 PS02 CO1 3 Image: Constraint of the state stane state of the state of the state of the state of th																
Strength of correlations (3:High, 2: Medium, 1:Low) PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO11 PO12 PS01 PS02 CO1 3 Image: Constant Signal									(L4)							
PO1PO2PO3PO4PO5PO6PO7PO8PO9PO10PO11PO12PS01PS02CO13IIIIIIIIIIICO23III <td></td> <td>(</td> <td>Contri</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Outco</td> <td>mes &</td> <td></td>		(Contri										Outco	mes &		
CO1 3 1 1 1 CO2 3 1 1 1 1 CO3 3 2 2 2 2 CO3 3 2		DO									-	,	DO12		DEOD	
CO23122CO3311332CO433322CO433322SyllabusUnit No.SyllabusUnit No.SyllabusMapped CO'sIntroduction to C: Introduction, Structure of C Program, A Simple C Program, C-Tokens, Basic Data types, Variables, Constants, Input / Output statements, Operators, Type conversion and Type casting. Conditional Branching Statements: if, if-else, if-else-if Statements and Switch case.Co1, CO22Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements. Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. Strings: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments.CO1, CO44Pointers: Declaration and Initialization of pointer variables, Pointer and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation.CO1, CO35User defined data-types: Introduction, bit-fields, Nested structures, ArrayCO1 CO3	CO1		I PU	2 PO3	P04	P05	PU0	PO/	PU8	P09	POIU	POIT	POIZ		1	
CO3 3 2 2 2 CO4 3 3 3 2 2 Syllabus Mapped CO's Unit No. Syllabus Mapped CO's 1 Introduction to C: Introduction, Structure of C Program, A Simple C Program, C-Tokens, Basic Data types, Variables, Constants, Input / Output statements, Operators, Type conversion and Type casting. CO1, CO2 Conditional Branching Statements: if, if-else, if-else-if Statements and Switch case. Conditional Branching Statements: if, if-else, if-else-if Statements and Switch case. Co1, CO2 2 Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements. Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. CO1, CO2 3 Functions: Introduction, Function declaration, Function definition and Function, String manipulation functions. CO1, CO4 4 Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. CO1, CO3 5 User defined data-types: Introduction, bit-fields, Nested structures, Array CO1, CO3		-													1	
CO4 3 3 3 2 2 Syllabus Mapped CO's Unit No. Syllabus Mapped CO's 1 Introduction to C: Introduction, Structure of C Program, A Simple C Program, C-Tokens, Basic Data types, Variables, Constants, Input / Output statements, Operators, Type conversion and Type casting. CO1, CO2 2 Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements. 2 Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements. 3 Functions: Introduction, String manipulation functions. 3 Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments. CO1, CO2 4 Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. CO1, CO3 5 User defined data-types: Introduction, bit-fields, Nested structures, Array																
Syllabus Unit No. Syllabus Mapped CO's 1 Introduction to C: Introduction, Structure of C Program, A Simple C Program, C-Tokens, Basic Data types, Variables, Constants, Input / Output statements, Operators, Type conversion and Type casting. CO1, CO2 2 Conditional Branching Statements: if, if-else, if-else-if Statements and Switch case. CO1, CO2 2 Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements. CO1, CO2 3 Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. CO1, CO2 3 Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments. CO1, CO4 4 Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. CO1, CO3 5 User defined data-types: Introduction, bit-fields, Nested structures, Array CO1, CO3		5	3							3	3					
Unit No.SyllabusMapped CO's1Introduction to C: Introduction, Structure of C Program, A Simple C Program, C-Tokens, Basic Data types, Variables, Constants, Input / Output statements, Operators, Type conversion and Type casting. Conditional Branching Statements: if, if-else, if-else-if Statements and Switch case.CO1, CO22Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements. Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. Strings: Introduction, String manipulation functions.CO1, CO23Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments.CO1, CO44Pointers: Declaration and Initialization of pointer variables, Pointer and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation.CO1, CO35User defined data-types: Introduction, bit-fields, Nested structures, ArrayCO1, CO3	001		5					Svl	abus	5	5					
1 Introduction to C: Introduction, Structure of C Program, A Simple C Program, C-Tokens, Basic Data types, Variables, Constants, Input / Output statements, Operators, Type conversion and Type casting. CO1, CO2 2 Conditional Branching Statements: if, if-else, if-else-if Statements and Switch case. CO1, CO2 2 Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements. CO1, CO2 Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. CO1, CO2 3 Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments. CO1, CO4 4 Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. CO1, CO3 5 User defined data-types: Introduction, bit-fields, Nested structures, Array CO1, CO3	Unit l											Mappe	d CO's			
statements, Operators, Type conversion and Type casting. Conditional Branching Statements: if, if-else, if-else-if Statements and Switch case.CO1, CO22Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements. Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. Strings: Introduction, String manipulation functions.CO1, CO23Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments.CO1, CO44Pointers: Declaration and Initialization of pointer variables, Pointers and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation.CO1, CO35User defined data-types: Introduction, bit-fields, Nested structures, ArrayCO1, CO3	1		Intro	luction	to C:	Intro	luction	, Stru	cture o	of C P	rogram,	A Sim	ple C			
Conditional Branching Statements: if, if-else, if-else-if Statements and Switch case.2Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements. Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. Strings: Introduction, String manipulation functions.CO1, CO23Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments.CO1, CO44Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation.CO1, CO35User defined data-types: Introduction, bit-fields, Nested structures, ArrayCO1, CO3			Program, C-Tokens, Basic Data types, Variables, Constants, Input / Output													
Switch case.Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements.CO1, CO2Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. Strings: Introduction, String manipulation functions.CO1, CO23Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments.CO1, CO44Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation.CO1, CO35User defined data-types: Introduction, bit-fields, Nested structures, ArrayCO1, CO3															CO1, CO2	
2 Iterative Statements: while, do-while and for loops, Nested loops, break and continue statements. CO1, CO2 Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. CO1, CO2 3 Functions: Introduction, String manipulation functions. CO1, CO4 4 Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. CO1, CO3 5 User defined data-types: Introduction, bit-fields, Nested structures, Array CO1, CO3																
and continue statements.CO1, CO2Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. Strings: Introduction, String manipulation functions.CO1, CO23Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments.CO1, CO44Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation.CO1, CO35User defined data-types: Introduction, bit-fields, Nested structures, ArrayCO1, CO3	2															
Arrays: Declaration, Accessing array elements, Storing values, Operations on arrays, Multi-dimensional arrays. Strings: Introduction, String manipulation functions.CO1, CO23Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments.CO1, CO44Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation.CO1, CO35User defined data-types: Introduction, bit-fields, Nested structures, ArrayCO1, CO3	2															
on arrays, Multi-dimensional arrays.Strings: Introduction, String manipulation functions.3Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments.CO1, CO44Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation.CO1, CO35User defined data-types: Introduction, bit-fields, Nested structures, ArrayCO1, CO3															CO^2	
Strings: Introduction, String manipulation functions.CO1, CO43Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments.CO1, CO44Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation.CO1, CO35User defined data-types: Introduction, bit-fields, Nested structures, ArrayCO1, CO3															1,002	
 Functions: Introduction, Function declaration, Function definition and Function call, Types of Functions, Parameter passing, Passing arrays to functions, Recursion, Storage classes, Command line arguments. Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation. User defined data-types: Introduction, bit-fields, Nested structures, Array 				•			-		on funct	ions.						
functions, Recursion, Storage classes, Command line arguments.4Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation.CO1, CO35User defined data-types: Introduction, bit-fields, Nested structures, ArrayCO1, CO3	3							_			nction of	definitio	n and			
 4 Pointers: Declaration and Initialization of pointer variables, Pointer arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation. 5 User defined data-types: Introduction, bit-fields, Nested structures, Array 															l, CO4	
 arithmetic, Pointers and arrays, Pointer to pointer, Array of pointers, Pointers and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation. 5 User defined data-types: Introduction, bit-fields, Nested structures, Array 											-					
 and functions, Dynamic memory allocation. Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation. 5 User defined data-types: Introduction, bit-fields, Nested structures, Array 	4		▲													
Pre-processor directives: The #define Directive, Undefining a Macro, Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation. CO1, CO3 5 User defined data-types: Introduction, bit-fields, Nested structures, Array CO1, CO3																
Token Pasting and Stringizing Operators, The #include Directive, Conditional Compilation. 5 User defined data-types: Introduction, bit-fields, Nested structures, Array															l, CO3	
Conditional Compilation. 5 User defined data-types: Introduction, bit-fields, Nested structures, Array CO1 CO3			-													
5 User defined data-types: Introduction, bit-fields, Nested structures, Array					-		ngiziilį	s Op	crai015,	The	πinciu		ccuve,			
	5						Introdu	iction	bit-fie	lds. Ne	ested str	uctures	Arrav			
			of structures, Structures and functions, Unions, enum, typedef.											CO	l, CO3	

PVP20

Department of Freshman Engineering

Files in C: Using Files in C, Read data from files, Writing data to files,									
Random access to files of records.									
Learning Resources									
Text Books									
1. Programming in C, ReemaThareja, AICTE Edition, 2018, Oxford University Press									
Reference Books									
1. Computer Science: A Structured Programming Approach Using C, B. A. Forouzan and R.F.									
Gilberg, Third Edition, 2007, Cengage Learning.									
2. Programming in C, PradipDey, Manas Ghosh, AICTE Edition, Oxford University Press.									
3. Programming with C, B. Gottfried, Third Edition, 2017, Schaum's outlines, McGraw Hill.									
4. Problem Solving & Program Design in C, Jeri R. Hanly, Ellot B. Koffman, 5th Edition, Pearson.									
e- Resources & other digital material									
1. http://cprogramminglanguage.net/									
2. https://www.geeksforgeeks.org/c-programming-language/									
3. https://www.greatlearning.in/academy/learn-for-free/courses/c-programming									
4. https://www.udemy.com/course/the-complete-c-programming/									
5. https://nptel.ac.in/courses/106/105/106105171/									

PVP20