|  |  |
| --- | --- |
| Course Code | CSI-301 |
| Course Title | Programming Fundamentals |
| Credit Hours | 4(3-1) |
| Prerequisite | - |
| Course Assessment | Total Marks = 80  Mid-term = 18, Sessional = 12, Final term = 30, Practical = 20. However sessional 12 marks are further distributed into 2 Assignments and 2 quizzes of each 3 marks. |
| Teaching Methodology | Lectures 64 Hours  Assignments 2, Quizzes 2 |
| Preamble | * To understand basic problem solving steps and logic constructs * To understand, Design and implement algorithms to solve real world problems |
| Course Contents | Introduction to problem solving, a brief review of Von-Neumann architecture, Introduction to programming, role of compiler and linker, introduction to algorithms, basic data types and variables, input/output constructs, arithmetic, comparison and logical operators, conditional statements and execution flow for conditional statements, repetitive statements and execution flow for repetitive statements, lists and their memory organization, multi-dimensional lists, introduction to modular programming, function definition and calling, stack rolling and unrolling, string and string operations, pointers/references, static and dynamic memory allocation, File I/O operations |
| Reference Book | 1. Starting out with Python, 4th Edition, Tony Gaddis.  2. Starting out with Programming Logic & Deigns, 4th Edition, Tony Gaddis,  3. The C Programming Language, 2nd Edition by Brian W. Kernighan, Dennis M. Ritchie  4. Object Oriented Programming in C++ by Robert Lafore  5. Introduction to Computation and Programming Using Python: With Application to Understanding Data, 2nd Edition by Guttag, John  6. Practice of Computing Using Python, 3rd Edition by William Punch & Richard Enbody  7. C How to Program, 7th Edition by Paul Deitel & Harvey Deitel  8. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman |
| Course Learning Outcomes | |  |  |  |  | | --- | --- | --- | --- | | CLOs# | Course Learning Outcomes | Domain | BT Level | | CLO-1 | Understand basic problem solving steps and logic constructs | Cognitive | 2 | | CLO-2 | Understand basic programing concepts | Cognitive | 2 | | CLO-3 | Apply basic programing concepts | Cognitive | 3 | | CLO-4 | Design and implement algorithms to solve real world problems. | Cognitive | 3 | |

|  |  |  |
| --- | --- | --- |
| Program Learning Outcomes | | |
| PLOs  S# | Program Learning Outcome | Computing Professional Graduate |
| PLO-01 | Academic Education | To prepare graduates as computing professionals |
| PLO-02 | Knowledge for Solving Computing Problems | Apply knowledge of computing fundamentals, knowledge of a computing specialization, and mathematics, science, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements |
| PLO-03 | Problem Analysis | Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines |
| PLO-04 | Design/ Development of Solutions | Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations |
| PLO-05 | Modern Tool Usage | Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations |
| PLO-06 | Individual and Team Work | Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings |
| PLO-07 | Communication | Communicate effectively with the computing community and with society at large about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions |
| PLO-08 | Computing Professionalism and Society | Understand and assess societal, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice |
| PLO-09 | Ethics | Understand and commit to professional ethics, responsibilities, and norms of professional computing practice |
| PLO-10 | Life-long Learning | Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mapping of CLOs with PLOs | | | | | | |
|  | CLO-1 | CLO-2 | CLO-3 | CLO-4 | CLO-5 | CLO-6 |
| PLO-01 | M | M | M | M | - | - |
| PLO-02 | S | S | S | S | - | - |
| PLO-03 | S | S | S | S | - | - |
| PLO-04 | S | S | S | S | - | - |
| PLO-05 | M | M | M | M | - | - |
| PLO-06 | L | L | L | L | - | - |
| PLO-07 | - | - | - | - | - | - |
| PLO-08 | - | - | - | - | - | - |
| PLO-09 | - | - | - | L | - | - |
| PLO-10 | - | - | - | - | - | - |

“S” =Strong ‘M” =Medium “L” =Low “- “=Not being Applicable

|  |  |  |
| --- | --- | --- |
| Lec# | Topic | Pages |
| 01 | a brief review of Von-Neumann architecture | 4-5 |
| 02 | Introduction to problem solving | 11 |
| 03 | Introduction to programming, role of compiler and linker | 12-13 |
| 04 | introduction to algorithms | 55-56 |
| 05 | basic data types and variables | 28-34 |
| 06 | input/output constructs | 24-28 |
| 07 | arithmetic, comparison | 34-38 |
| 08 | logical operators | 38-40 |
| 09 | conditional statements and execution flow for conditional statements | 58-60 |
| 10 | repetitive statements and execution flow for repetitive statements, while loop | 64-70 |
| 11 | repetitive statements and execution flow for repetitive statements, while loop | 80-99 |
| 12 | repetitive statements and execution flow for repetitive statements, for loop | 100-103 |
| 13 | Switch multiple statements | 109-109 |
| 14 | Do-while loop | 113-114 |
| 15 | Break continue statement | 114-115 |
| 16 | introduction to modular programming | 141-142 |
| 17 | function definition and calling, | 145-148 |
| 18 | Calling function by value by reference | 152-167 |
| 19 | stack rolling and unrolling | 151 |
| 20 | Random number generation | 153 |
| 21 | Recursion | 167-170 |
| 22 | lists and their memory organization | 196-198 |
| 23 | multi-dimensional lists | 229-232 |
| 24 | Passing array to function | 212-214 |
| 25 | MID TERM EXAM |  |
| 26 | Searching array | 223-224 |
| 27 | pointers | 254-255 |
| 28 | Pointer and references | 257-261 |
| 29 | Relationship between pointer and array | 275-276 |
| 30 | Array of pointers | 280-282 |
| 31 | Pointers to function | 285-290 |
| 32 | Introduction to string | 310-311 |
| 33 | String library functions | 312-313 |
| 34 | String conversion functions | 317-318 |
| 35 | Standard i/o library functions | 322-324 |
| 36 | String manipulation functions | 326-327 |
| 37 | String handling library | 331-341 |
| 38 | String handling library | 331-341 |
| 39 | String handling library | 331-341 |
| 40 | Search functions of string handling library | 331-341 |
| 41 | static and dynamic memory allocation |  |
| 42 | Introduction to file | 418-420 |
| 43 | Files and streams | 420 |
| 44 | Sequential access file | 426 |
| 45 | Random access file | 430 |
| 46 | File operations | 437 |
| 47 | PRACTICAL |  |
| 48 | FINAL TERMA EXAM |  |