**Course Information**

**Program :** B.Sc. Engineering in CSE

**Course Code :** CSE 111

**Course Title :** Structured Programming Language

**Course Credit :** 3.00

**Contact Hours :** 3hrs

**Semester :** Spring 2020

**Intake :** 44th [Shift: Day]

**Section :** 13

**Prerequisites :** CSE 101 Introduction to Computers

**Course Objectives**

The course “*Structured programming language*” has been designed to focus on to learn the fundamental concepts of procedural programming in C. This course also focusses on to introduce to the student’s basic problem solving techniques and necessary programming tools to solve them.

**Course Synopsis**

This course has been structured in such a way that student will learn from the very fundamentals of programming right through to the basic loops, operators, functions, recursion techniques, one dimensional and multi-dimensional arrays, structure, pointers, and file input output operation. It also helps the students to become familiarize how to make a small but complete project using C language.

**Assessment**

|  |  |  |
| --- | --- | --- |
| Class Participation | : | 10% |
| Assignment/Presentation | : | 10% |
| Class Test | : | 10% |
| Midterm Examination | : | 30% |
| Final Examination | : | 40% |



**Course Outcomes (COs)**

After completion of this course students will be able to:

|  |  |
| --- | --- |
| **CO1:** | **Describe** the objective of programming concept and terms related to structured programming languages. |
| **CO2:** | **Understand** the real life problems and find solutions. |
| **CO3:** | **Apply** programming skillsindividual or in group for different problem analysis and solutions. |
| **CO4:** | **Analyze** logical and mathematical expressions for developing a better software solution. |

**Mapping of Course Outcomes (COs) to Program Outcomes (POs)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
| CO1 | √ |  |  |  |  |  |  |  |  |  |  |  |
| CO2 |  | √ |  |  |  |  |  |  |  |  |  |  |
| CO3 |  |  | √ |  |  |  |  |  |  |  |  |  |
| CO4 |  |  |  | √ |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl. No. | COs | Corresponding POs | Bloom’s taxonomy domain/level | Delivery methods and activities | Assessment tools |
| 1 | **Describe** the objective of programming concept and terms related to structured programming languages. | **PO1** | Remember | Class Lecture and Discussion | Class test, Assignment, and Midterm exam |
| 2 | **Understand** the real life problems and find solutions. | **PO2** | Understand | Class Lecture and Discussion | Class test, Assignment, and Midterm Exam |
| 3 | **Apply** programming skillsindividual or in group for different problem analysis and solutions. | **PO3** | Apply | Class Lecture and Discussion | Class test, Assignment, Midterm, and Final Exam |
| 4 | **Analyze** logical and mathematical expressions for developing a better software solution. | **PO4** | Analyze | Class Lecture and Discussion | Class test, Assignment, and Final Exam |

**Descriptions of Program Outcomes (POs)**

|  |  |
| --- | --- |
| PO1 | **Engineering Knowledge (Cognitive):** Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems. |
| PO2 | **Problem Analysis (Cognitive):** Identify, formulate, research the literature and analyze complex engineering problems and reach substantiated conclusions using first principles of mathematics, the natural sciences and the engineering sciences. |
| PO3 | **Design/Development of Solutions (Cognitive, Affective):** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety as well as cultural, societal and environmental concerns. |
| PO4 | **Investigation (Cognitive, Psychomotor):** Conduct investigations of complex problems, considering design of experiments, analysis and interpretation of data and synthesis of information to provide valid conclusions. |
| PO5 | **Modern Tool Usage (Psychomotor, Cognitive):** 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. |
| PO6 | **The Engineer and Society (Affective):** Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice. |
| PO7 | **Environment and Sustainability (Affective, Cognitive):** Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development. |
| PO8 | **Ethics (Affective):** Apply ethical principles and commit to professional ethics, responsibilities and the norms of the engineering practice. |
| PO9 | **Individual Work and Teamwork (Psychomotor, Affective):** Function effectively as an individual and as a member or leader of diverse teams as well as in multidisciplinary settings. |
| PO10 | **Communication (Psychomotor, Affective):** Communicate effectively about complex engineering activities with the engineering community and with society at large. Be able to comprehend and write effective reports, design documentation, make effective presentations and give and receive clear instructions. |
| PO11 | **Project Management and Finance (Cognitive, Psychomotor):** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work as a member or a leader of a team to manage projects in multidisciplinary environments. |
| PO12 | **Life-Long Learning (Affective, Psychomotor):** Recognize the need for and have the preparation and ability to engage in independent, life-long learning in the broadest context of technological change. |

**Weekly Schedule**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Lecture** | **Topics** | **Chapter** | **CO** | **Exam**  **(Mark)** |
| 1 | 1 | C Introduction: Character Set, C Tokens, Keywords and Identifiers | 1 | **CO1** | **Mid- Term**  **Exam(30)** |
|  | 2 | Constants and Variables, Data Types | 4 | **CO1** |
|  | 3 | Operators and Expressions: Arithmetic & logical operators, Operators Precedence and Associativity | 4 | **CO1**  **+**  **CO2** |
| 2 | 4 | Type Conversion in Expressions, Conditional Operator, Library Functions | 4 | **CO1**  **+**  **CO2** |
|  | 5 | printf, scanf, getchar and putchar example | 2 | **CO1** |
|  | 6 | Control Statements: The if-else Statement, if ladder, nesting of if-else | 3 | **CO2** |
| 3 | 7 | The while Statement, The do-while Statement, The for Statement | 3 | **CO1**  **+**  **CO3** |
|  | 8 | Nested Loop | “ | **CO3** |
|  | 9 | Nested Loop | “ | “ |
| 4 | 10 | Loop example & problem analysis | 3 | **CO2+**  **CO3** |
|  | 11 | The switch Statement, break Statement, |  | “ |
|  | 12 | continue and goto Statement, Comma Operator |  | “ |
| 5 | 13 | Function Prototypes, Function call | 7 | **CO2**  **+**  **CO3** |
|  | 14 | Passing Arguments to a Function, Return statement | “ | “ |
|  | 15 | Function Call with different Parameter | 6 | “ |
| 6 | 16 | Passing Arguments to a Function, Return statement | “ | “ |
|  | 17 | ACM-ICPC: Judging Systems & UVA problem solving | “ | “ |
|  | 18 | Troubleshooting Case Study and Review classes |  |  |
| 7 |  | **Mid Term Examination** |  |  | **Term Final**  **Exam**  **(40)** |
| 8 | 19 | UVA problem solving |  | **CO3+CO4** |
|  | 20 | Arrays: Declaration and initialization array, Traversing array | 5 | “ |
|  | 21 | Passing Arrays to Functions, Operation that can be performed on array | “ | “ |
| 9 | 22 | Multidimensional Arrays: declaration, initialization and accessing multidimensional array. | 5 | **CO3** |
|  | 23 | Matrix operations & implementation | “ | “ |
|  | 24 | String: Arrays and Strings | “ | “ |
| 10 | 25 | Recursion: recursive functions, Types of recursion: direct recursion | 7 | **CO2**  **+**  **CO3** |
|  | 26 | indirect recursion and tail recursion | “ | “ |
|  | 27 | Pointers: Fundamentals, Pointer Declarations, Passing Pointers to Functions, | 6 | “ |
| 11 | 28 | Pointers and One-Dimensional Arrays, Pointers and strings, String Operations | 6 | **CO3**  **+**  **CO4** |
|  | 29 | Dynamic Memory Allocation, Arrays of Pointers | “ | “ |
|  | 30 | Structures: Defining a Structure, Processing a Structure | 10 | “ |
| 12 | 31 | Unions: declare a union, accessing a member, initialize | “ | **CO3**  **+**  **CO4** |
|  | 32 | User-Defined Data Types (typedef), Structures and Pointers | 11 | “ |
|  | 33 | Declare a file, open, process and close a file | 9 | “ |
| 13 | 34 | Read data from file: fscanf(), fgets(), fgets(), fread() | 9 | **CO3**  **+**  **CO4** |
|  | 35 | File operations & example | “ | “ |
|  | 36 | Troubleshooting Case Study and Review classes | “ | “ |

**Description of Cognitive Domain( Anderson & Krathwohl’s Taxonomy 2001)**

The cognitive domain involves the development of our mental skills and the acquisition of knowledge.

|  |  |  |  |
| --- | --- | --- | --- |
| Level | Category | Meaning | Keywords |
| C1 | Remembering | Recognizing or recalling knowledge from memory.  Remembering is when memory is used to produce or retrieve definitions, facts, or lists, or to recite previously learned information. | Define, describe, Explain, Identify, label, list, match, name, quote, recall, recite, tell, write, Demonstrate, State, Show, Summarize |
| C2 | Understanding | Constructing meaning from different types of functions be they written or graphic messages or activities like interpreting, exemplifying, | Classify, compare, exemplify, conclude, discuss, describe, explain, identify, paraphrase, Show, Brief |
|  |  | classifying, summarizing, inferring, comparing, or explaining. |  |
| C3 | Applying | Carrying out or using a procedure through executing, or implementing. Applying relates to or refers to situations where learned material is used through products like models, presentations, interviews or simulations. | Apply, Build, change, compute, implement, prepare, produce, role play, select, show, transfer, use  Produce, Generate, Construct, Write, Create, Find |
| C4 | Analyzing | Breaking materials or concepts into parts, determining how the parts relate to one another or how they interrelate, or how the parts relate to an overall structure or purpose. Mental actions included in this function are differentiating, organizing, and attributing, as well as being able to distinguish between the components or parts. When one is analyzing, he/she can illustrate this mental function by creating spreadsheets, surveys, charts, or diagrams, or graphic representations. | Analyze, characterize, classify, compare, contrast, debate, deconstruct, deduce, differentiate, discriminate, distinguish, examine, Explain, Evaluate, Find, organize, outline, output relate, research, separate, structure, predict, report |
| C5 | Evaluating | Making judgments based on criteria and standards through checking and critiquing.  Critiques, recommendations, and reports are some of the products that can be created to demonstrate the processes of evaluation. | Appraise, argue, assess, choose, conclude, critique, decide, evaluate, judge, justify, predict, prioritize, prove, rank, rate, select, monitor |
| C6 | Creating | Putting elements together to form a coherent or functional whole; reorganizing elements into a new pattern or structure through generating, planning, or producing. Creating requires users to put parts together in a new way, or synthesize parts into something new and  different creating a new form or product. This process is the most difficult mental function. | Construct, design, develop, generate, hypothesize, invent, plan, produce, compose, create, make, perform, plan, produce |

**Teaching materials**

**Required References:**

**1. Teach Yourself C**- Herbert Schildt

**2. Programming in C**- E. Balagurushamy

**3. Let us c**- Yashwant kanetkar

**Recommended References:**

1. **Programming with C**- Schaums Outlines – Byron Gottfried

2. **Introduction to C Programming**- Reema Thareja

**Other materials:** Lecture Notes, Web links reference, etc. [if any. do yourself]

**Overall Assessment Scheme**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Assessment Area** | **CO** | | |  | **Assessment Area mark** |
|  | **CO1** | **CO2** | **CO3** | **CO4** |  |
| **Mid Term** | 10 | 10 | 10 |  | 30 |
| **Final** | 10 | 10 | 10 | 10 | 40 |

**Grading System**

|  |  |  |  |
| --- | --- | --- | --- |
| **Numerical Grade** | **Letter Grade** | | **Grade Pont** |
| 80% and above | A+ | (A Plus) | 4.00 |
| 75% to less than 80% | A | (A Regular) | 3.75 |
| 70% to less than 75% | A- | (A Minus) | 3.50 |
| 65% to less than 70% | B+ | (B Plus) | 3.25 |
| 60% to less than 65% | B | (B Regular) | 3.00 |
| 55% to less than 60% | B- | (B Minus) | 2.75 |
| 50% to less than 55% | C+ | (C Plus) | 2.50 |
| 45% to less than 50% | C | (C Regular) | 2.25 |
| 40% to less than 45% | D |  | 2.00 |
| Less than 40% | F |  | 0.00 |

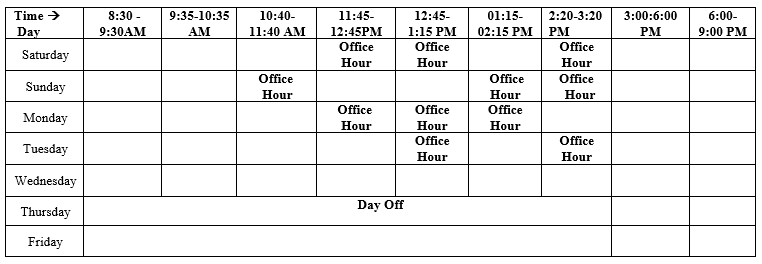
**Instructor Information**

|  |  |
| --- | --- |
| **Instructor :** | **Md. Anwar Husssen Wadud**  Lecturer,  Department of Computer Science & Engineering |
| **Office :** | Room No-516 (Permanent Campus) |
| **Phone :** | +8801515 - 607055 |
| **Email :** | anwarcse28@gmail.com |

**Class Schedule**

|  |  |  |
| --- | --- | --- |
| **Day** | **Time** | **Room No** |
| Saturday | 01:15 PM – 02:15 PM | 320 |
| Tuesday | 09:35 AM – 11:40 AM | 319 |

**Office Hours**



**Special Instructions**

* Students are expected to attend all classes and examinations. A student MUST have at least

70% class attendance to sit for the final exam.

* Students will not be allowed to enter into the classroom after 20 minutes of the starting time.
* For plagiarism, the grade will automatically become zero for that exam/assignment.
* All mobile phones MUST be turned to silent mode during class and exam period.
* There is zero tolerance for cheating in exam. The only penalty for cheating is expulsion

for several semesters as decided by the Disciplinary Committee of the university.

. . . . . . . [do yourself]

|  |  |  |
| --- | --- | --- |
| **Prepared by:** | **Checked by:** | **Approved by:** |