**FACULTY OF COMPUTER AND MATHEMATICAL SCIENCES**

**SCHEME OF WORK**

**Course : Fundamentals of Computer Problem Solving**

**Course Code : CSC128**

| **Week** | **Hour(s)** | **Topic** | **Activity** |
| --- | --- | --- | --- |
| 1 | 2 | **Topic 1: Introduction to Computer, Program and Programming Language**   * Introduction to Programming   + Definition of computer and computer program   + Importance of computer programming   + Importance of good programs.   + Relationship between compilers, interpreters, assemblers and programs | Lecture |
| 3 | **Topic 1: Introduction(cont’d)**   * Program Development Life Cycle – Problem analysis, design, implementation, testing, maintenance (documentation) | Lecture & Tutorial |
| 2 | 3 | **Topic 2: Basic Elements of Computer Program**   * Identifiers, variables, constants, and reserved words * Rules for naming identifiers * Declaring data variables and constants * Basic data types – integer, float, double, bool, char (including sequence of characters **without** using the string keyword) * Arithmetic operators – addition, subtraction, multiplication, division, modulus (**excluding** pre-increment operator and pre- decrement operators ) * Arithmetic expression – unary and binary operands * Operators precedence | Lecture & Tutorial  **Quiz 1**  **Assignment 1** |
| 2 | **Lab Session**   * Introduction to C++ Integrated Development Environment (IDE) * Compile and execute simple program * Debug and execute given programs | * Announcement of project (a group of two members) * Rubrics and criteria of evaluation |
| 3 | 3 | **Topic 2: Basic Elements of Computer**  **Program (cont’d)**   * Assignment statement (excluding compound assignment statement such as += etc.) * Assign (copy) a string (strcpy()). * Introduce some mathematical library functions **–** sqrt(), abs(), pow(), etc. * Input/output statement (avoid arithmetic expression in output) * String input statement (gets(), getline(), etc.) * C++ program structure * Programming process, debugging and error handling * Four types of control structure **–** sequential, selection, repetition/looping , modular | Lecture & Tutorial |
| 2 | **Lab Session**   * Complete a source code base on problem that is being discussed * Convert from a flowchart and pseudocode given to a program to get an output * Debug and execute a given program (arithmetic expression) | Project proposal discussion and submission |
| 4 | 2 | **Topic 3 : Selection Control Structure**   * + Relational and logical operators   + Boolean expression and Boolean value   + Compound Boolean Expression   + Precedence of operators | Lecture & Tutorial  **Quiz 2** |
| 3 | **Lab Session**   * Debug and execute a given program (arithmetic expression) * Solve a given problem by applying arithmetic expressions | * Progress report of project (stage 1: Analysis, Input, Output, Process, Initial design of algorithm, source code) |
| 5 | 3 | **Topic 3 : Selection Control Structure (cont’d)**   * + Types of selection Control Structures     - * One way selection       * Two ways selection       * Multiple selection | Lecture & Tutorial |
| 2 | **Lab Session**   * Debug and execute a given program (selection control structures) * apply selection statement in decision making problem * enhance the usage of the formatting output symbol |  |
| 6 | 3 | * nested selection * switch statement * string compare (strcmp()) | Lecture & Tutorial  **Quiz 3** |
| 2 | **Lab Session**   * Debug and execute a given program (nested selection ) * Explain the output based on the various input from the user * Solve problem using strcmp() function * Apply the nested selection to a given problem | * Progress report of project (stage 2: Algorithm refinement including **selection structure**, source code) |
| 7 | 3 | **Topic 4 : Repetition Control Structure**   * Requirements – initialization, evaluation (condition) and updating * Types of repetition structure   + - * Counter controlled loop (while, do..while,for) | Lecture & Tutorial |
| 2 | **Lab Session**   * Debug and execute a given program (repetition control structures) * Explain the output from the source code given * Solve repetition problem using counter structure |  |
| 8 | 3 | **Topic 4 : Repetition Control Structure (cont’d)**   * Types of repetition structure   + - * Sentinel controlled loop (while, do..while)       * Flag controlled loop (Boolean variables) (while, do..while,for) | Lecture & Tutorial  **Assignment 2** |
| 2 | **Lab Session**   * Debug and execute a given program (repetition control structures) * Solve iteration problems using sentinel and flag structures |  |
| 9 | 3 | **Topic 4 : Repetition Control Structure (cont’d)**   * nested loop (for statement) * continue and break statement | Lecture & Tutorial  **Quiz 4** |
| 2 | **Lab Session**   * Debug and execute a given program (repetition control structures) * Solve nested loop problems using for loop statements only | Progress report of project (stage 3: Algorithm refinement including **repetition structure**, source code) |
| 10 | 3 | **Topic 5: Function**   * Introduction to function * Predefined function – introduce some common mathematical functions   sqrt(), abs(), pow(), setw(), setprecision()   * Types of variable and its scope – block, local & global * User-defined function   + - * Function prototype       * Function definition (header and body)       * Function invocation | Lecture & Tutorial |
| 2 | **Lab Session**   * Debug and execute sample program using predefined function * Solve a problem using predefined function |  |
| 11 | 3 | **Topic 5: Function (cont’d)**   * Types of parameter – actual and formal * Parameter passing   + - * Without parameter       * With parameter (by value) | Lecture & Tutorial |
| 2 | **Lab Session**   * Debug and execute sample program using user-defined function * Solve a problem using user-defined function with parameter passing by value |  |
| 12 | 3 | **Topic 5: Function (cont’d)**   * Parameter passing (cont’d)   + - * With parameter (by reference) | Lecture & Tutorial |
| 2 | **Lab Session**   * Debug and execute sample program using user-defined function * Solve a problem using user-defined function with parameter passing by reference |  |
| 13 | 3 | **Topic 5: Function (cont’d)**   * Return value   + - * By value       * By reference | Lecture & Tutorial  **Quiz 5** |
| 2 | **Lab Session**   * Debug and execute sample program using user-defined function * Solve a problem using user-defined function return by value and by reference | Progress report of project (stage 4: Complete algorithm design including **function structure**, complete source code) |
| 14 | 5 | **Incremental Project Presentation** | * Group presentation * Complete report submission including analysis, algorithm, complete program (meaningful identifiers, correct indentation, with appropriate comments) and samples of input and output. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Assessment** |  |  |  |
| Final Examination | : | 3-hour paper | 50 % |
| Course Work | : | Tests(2)  Quizzes (5)  Assignments (2)  Incremental Project  Total | 50%   * 20% * 15 % * 5% * 10%   100 % |
|  |  |  |  |

**Recommended Text Book:**

Liang, Y.D., *Introduction to Programming with C++,* 2nd edition, Pearson Higher Education, 2010.

**References**

1. Malik, D.S., *C++ Programming: From Problem Analysis To Program Design*, 6th edition, Course Technology, 2013.

2. Farrell, Joyce, *Programming Logic and Design Comprehensive,* 2nd edition, Course Technology 2002.

3. Zak, Dianne, *An Introduction to Programming with C++*, 2nd edition, Course Technology, 2001.

4. Jamal Othman, *Fundamentals of Programming : With Examples in C, C++ and Java*, 1st edition, UPENA, 2010.