|  |  |
| --- | --- |
| **College**  | Engineering and Technology |
| **Department** | Computer Systems Engineering |
| **Program** | Bachelor of Computer Systems Engineering |
| **Course Title** | Computer Programming | **Course Number:** | 12140101 |
| **Year** | 2020 / 2021 | **Semester:** | Second |
| **Prerequisite(s)** | -- |
| **Instructor** | Mohamed Khalil |
| **Instructor's e-mail** | m.khalil@ptuk.edu.ps |
| **Office Hours** | Sunday : 9:00 -13:00 |
| **Class Time** | **Sections 01:** Sun, Tue, Thu (08 – 09)  | **Class Room:** | H 108 |
| **Course description** | This course introduces the student to C++ programming through a study of the concepts of program specification and design, algorithm development, and coding and testing using a modern software development environment using C++. Students learn how to write programs in a high level programming language (C++). Topics covered include fundamentals of algorithms, flowcharts, problem solving, programming concepts, control structures, arrays, and pointers. Throughout the semester, problem solving skills will be stressed and applied to solving computing problems. Weekly laboratory experiments will provide hands-on experience in topics covered in this course. |
| **Course Intended Learning Outcomes (CILOs)** | 1. **Knowledge and understanding:**
	1. Have an understanding of the main programming constructs of C++.
	2. Have an understanding of the role of design in the development of programming solutions to problems.
	3. Have knowledge of some standard algorithms and data structures.
	4. To become familiar with different types of’ programming languages.
	5. To understand a typical C++ program-development environment.
	6. To be able to write simple computer programs in C++.
	7. To be able to use simple Pointers.
	8. To become familiar with fundamental data types.
	9. To be able to use Structures.
	10. To be able to write Classes.
2. **Intellectual / Cognitive skills:**
	1. Develop the ability to analyze problems and propose algorithms to solve them.
	2. Appreciate the subtleties related to C++ programming language.
	3. Appreciate the function component design and its usage.
	4. Use the numerical methods in C++ for many engineering applications.
3. **Subject specialization and practical skills:**
	1. Be able to write computer programs to solve practical engineering problems.
	2. Be able to design efficient computer programs to solve practical engineering problems.
	3. Design: write C++ code for different applications.
	4. Testing the looping function.
	5. Viewing the example results.
	6. Use the lab for many C++ programs in different applications.
4. **General and transferable skills:**
	1. Deploy communication skills.
	2. Deploy research skills.
	3. Working effectively within a group to analyze and writing C++ programs.
 |
| **Textbook(s)** | 1. C++ How to Program, 7th edition, P. Deitel and 11. Deitel, Prentice Hall, 2009.
2. C++ Programming: From problem analysis to program design, 6th edition, D.S. Malik, Thomson Learning, 2011.
 |
| **Other required material (References):** | 1. Richard Halterman, “Fundamentals of Programming: An Introduction to Computer
2. Programming Using C++” 1995 - Jofel Adams, Sanford Leestma, and Larry Nyhoff, “TurboC++: An introduction to computing” Prentice-Hall.
 |
| **Other Resources used (e.g. e-learning, field visits, periodicals, software, etc. )** | 1. http://www.cpIuspIus.com/ Textbook website.
2. http://www.deitel.com/books/cpphtp5/ Free C and C++ resources.
3. http://www.freeprogrammingresources.com/freetutr.htmt
4. PTUK e-learning management system (<https://lms.ptuk.edu.ps>)
5. Visual studio
 |

|  |
| --- |
| **Course Teaching Methods** |
| **Teaching Method** | **CILOs** |
| 1. Recorded Lectures
 | 1 , 2 |
| 1. Regular Lectures
 | 2, 3 |
| 1. Lab Sessions
 | 3, 4 |
| 1. Reading Materials
 | 1 |
| 1. Assignments
 | 3, 4 |
| 1. Case Studies
 | 2, 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Assessment Type** | **Details/Explanation of assessment in relation to CILOs** | **Weight** | **Date(s)** |
| **Midterm Exam**  | 1, 2 | 20 | Week: 06  |
| **Assignments (02)** |  | 20 | Week: 6, week 10 |
| **Final Exam** | 1, 2 | 30 | Week: 15-16 |
| **Total** |  | 70 | -- |
| **Laboratory/Practical**  | 3, 4 | 30 | Weeks: 01-14 |
|  |  | 100 |  |
|  |  |  |  |

|  |
| --- |
| **Course Intended Learning Outcomes (CILOs)** |
| **CILOs** | **Mapping to Program ILOs** |
| **On successful completion of the course, students will be able to:** | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **I** | **j** |
| ILO 1 (mentioned above) |  |  |  |  |  |  |  |  |  |  |
| ILO 2 (mentioned above) |  |  |  |  |  |  |  |  |  |  |
| ILO 3 (mentioned above) |  |  |  |  |  |  |  |  |  |  |
| ILO 4 (mentioned above) |  |  |  |  |  |  |  |  |  |  |

|  |
| --- |
| **Course Weekly Breakdown** |
| **Week** | **Date** | **Topics Covered** | **CILO** | **Lab Activities** | **Assessment** |
| 01 | 20-24/09/2020 | **Basic Elements of C++:**1. Basics of a C++ program.
2. Data types and Variables.
3. Arithmetic Operators, operator precedence, Expressions.
4. Type Conversion (Casting).
 |  | 1. Introduction to using the C++ compiler and start editing and running programs
2. Writing our first program
 |  |
| 02 | 27/9-1/10/2020 | 1. Variables and Assignment statements.
2. Increment and decrement operators.
3. Basic input and output.
4. Preprocessor directives.
 |  | Basic Input and outputConditional Statements |  |
| 03 | 4-8/10/2020 | **Control Structure (selection):**1. Relational operators
2. Logical operators and logical expression
3. Selection (if and if... else).

  |  | Selections structures |  |
| 04 | 11-15/10/2020 | 1. The conditional operator (?:)
2. The switch statement.
 |  | Selections structures |  |
| 05 | 18-22/10/2020 | **Control Structure (repetition):**1. The while loop
2. The for loop.
3. The do … while loop.

  |  | Repetition Statements |  |
| 06 | 25-29/10/2020 | 1. Nested control structures.
2. Break and continue statements.
 |  | Repetition Statements |  |
| 07 | 1-5/11/2020 | **User defined functions:**1. Predefined functions.
2. User-defined functions.
3. Value-returning and void functions.
4. Value and reference parameters, memory allocation
 |  | Functions |  |
| 08 | 8-12/11/2020 | 1. Function overloading
2. Default arguments
3. Scope of an identifier
4. Global variables, and side effects
5. Recursion
 |  | Functions |  |
| 09 | 15-19/11/2020 | **Arrays:**1. Accessing arrays components

Processing one-dimensional arrays |  | Arrays |  |
| 10 | 22-26/11/2020 | **Arrays:**1. Array index and bounds.
2. Array initialization during declaration.
3. Restriction on arrays processing.
4. Array as function parameter.
5. 2D array processing.
 |  | Arrays | Assignment#3 |
| 11 | 29/11-3/12/2020 | **Pointers:** 1. Introduction.
2. Pointer variable Declarations and Initialization.
 |  | Pointers |  |
| 12 | 6-10/12/2020 | 1. Pointer Operators.
2. Passing Arguments to Functions by Reference with Pointers.
3. Relationship between pointers and arrays
 |  | Pointers |  |
| 13 | 13-17/12/2020 | **Strings:**1. String Type.
2. String operations: length; size; operator (+); find; substr; swap; and concatenation.
 |  | Strings |  |
| 14 | 20-24/12/2020 | **Introduction to OOP** 1. Defining classes and objects
 |   | Classes and Objects |  |
|  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Prepared by:** | **Mohammad Khalil** | **Signature:** |  |
| **Head of Department:** | **Dr. Mohammad Khalil** | **Signature:** |  |
| **Date:** |  |