MULTIMEDIA UNIVERSITY®

## COURSE INFORMATION

| | | |
|---|---|---|
| 1 . | **Name of Course** | Concepts of Programming Languages |
| 2 . | **Course Code** | TCP2651 |
| 3 . | **Type of Course**<br>**(e.g. : Core, major, elective etc.)** | Elective |
| 4 . | **Synopsis** | 1. Increased capacity to express ideas.  The study of programming language concepts builds an appreciation for valuable language features and constructs and encourages programmers to use them.<br>2. Improved background for choosing appropriate language.<br>3. Increased ability to learn new languages. It is essential that practising programmers know the vocabulary and fundamental concepts of programming languages so they can read and understand programming language descriptions<br>4. Overall advancement of computing. |
| 5 . | **Version**<br>**(State the date of theSenate's approval - previous and the current approval date)** | Current: January 2018<br>Previous: June 2016 |
| 6 . | **Name(s) of Academic Staff** | Ku Day Chyi |
| 7 . | **Semester and Year Offered** | Trimester 2 (Gamma level) |
| 8 . | **Credit Value** | 4 |
| 9 . | **Pre-Requisite** | TCP1201 Object-Oriented Programming and Data Structure |
| 10 . | **Objective of the course in the programme:**<br>To expose students to the fundamental concepts of programming languages by discussing the design issues of the various language constructs, examining the design choices for these constructs in some of the common languages, and critically comparing design alternatives. This is essential for students to quickly grasp new programming languages. | |
| 11 . | **Justification for including the course in the programme:**<br>This subject is the foundation for the students to easily grasp any new programming languages. | |

| | | | |
|---|---|---|---|
| 12 . | **Course Learning Outcomes (CLO)** | **Domain** | **Level** |
| | **CLO1:** Identify the concepts of structures of programming languages correctly. | Cognitive | 1 |
| | **CLO2:** Demonstrate different language implementations correctly and effectively. | Cognitive | 3 |
| | **CLO3:** Distinguish how design issues affect the build of  programming language constructs. | Cognitive | 2 |
| | | | |

13 . **Mapping of the Course Learning Outcomes to the Programme Learning Outcomes, Teaching Methods and Assessment:**

| **Course Learning Outcomes (CLO) (Must tally with CLOs in item 12)** | **Programme Learning Outcomes (PLO)** | | | | | | | | | | | | **Teaching Methods** | **Assessment Method** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PLO1 | PLO2 | PLO3 | PLO4 | PLO5 | PLO6 | PLO7 | PLO8 | PLO9 | PLO10 | PLO11 | PLO12 | | |
| CLO1 | | ✓ | ✓ | ✓ | | | | | | | | | Lecture/Tutorial | Final Exam/Test/Assignment |
| CLO2 | | ✓ | ✓ | ✓ | | | | | | | | | Lecture/Tutorial | Final Exam/Test/Assignment |
| CLO3 | | ✓ | ✓ | ✓ | | | | | | | | | Lecture/Tutorial | Final Exam/Test/Assignment |
| | | | | | | | | | | | | | | |
| **Total** | | **3** | **3** | **3** | | | | | | | | | *Indicate the relevancy between the CLO and PLO by ticking "✓" the appropriate relevant box (This description must be read together with standards 2.1.2, 2.2.1, and 2.2.2 in Area 2 – pages 16 & 18 of COPPA 2.0)* | |

14 . **Transferable Skills:**

Time management: Develop through preparation and completing the assignment and presentation, final exam and test. Asssement via assignment report and presentation, final exam and

15 . **Distribution of Student Learning Time (SLT)**

| | **Course Content Outline** | ****CLO** | **Teaching and Learning Activities**<br>**Guided Learning (F2F)*** | | | | **Guided Learning (NF2F)*** | **Independent Learning (NF2F)*** | **Total SLT** |
|---|---|---|---|---|---|---|---|---|---|
| | | | ***L** | ***T** | ***P** | ***O** | | | |
| **1** | **Introduction**<br>Programming domains; programming paradigms; language evaluation criteria; language implementation methods. | | 2 | 2 | | | | 4 | 8 |
| **2** | **Formal Description of Syntax**<br>Lexems, syntax, semantics, grammar definitions; context-free grammar (BNF, EBNF); syntax graph; parse trees; ambiguous grammar; left and right derivation; operator precedence and associativity. | | 4 | 4 | | | | 8 | 16 |
| **3** | **Grammar and Semantics**<br>Problems of context-free grammar; attribute grammar; fully attributed parse tree; dynamic semantics – informal semantics, translational semantics, operational semantics, denotational semantics. | | 4 | 2 | | | 4 | 6 | 16 |
| **4** | **Names, Bindings, Type Checking, and Scopes**<br>Names; variables; bindings – type binding (static and dynamic type binding), storage binding (static and dynamic allocations), scope binding (static and dynamic scope bindings); type checking. | | 4 | 2 | | | 4 | 6 | 16 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 5 | **Data Types**<br>Data types; enumeration types; array types; record types; pointer types. | | 2 | 4 | | | 4 | 6 | 16 |
| 6 | **Expressions and Assignment Statements**<br>Expressions; operators; evaluation order; type conversions; type of expressions; short-circuit evaluation. | | 4 | 4 | | | | 8 | 16 |
| 7 | **Statement-Level Control Structures**<br>Control structures; selection statements; iterative statements; unconditional branching. | | 2 | 4 | | | 4 | 6 | 16 |
| 8 | **Subprograms**<br>Control structures; selection statements; iterative statements; unconditional branching; design issues for subprograms. | | 2 | 2 | | | | 4 | 8 |
| | | | | | | | | **Total SLT** | **112** |

**SUMMATIVE ASSESSMENT**

| | | | |
|---|---|---|---|
| **1. Continuous Assessment** | **Percentage %** | **Total SLT** | |
| Assignment | 20% | 21 | |
| Test | 20% | 5 | |
| | | | |
| | | | |
| **Total SLT for Continuous Assessment** | | **26** | |
| | | | |
| **2. Final Assessment** | **Percentage %** | **Total SLT** | |
| | | **F2F** | **ILT** |
| Final Exam | 60% | 2 | 20 |
| **Total SLT for Final Assessment (F2F + NF2F)** | | **22** | |
| | | | |
| **Grand Total** | **100%** | **160** | |

****Indicate the CLO based on the CLO's numbering in Item 12.**
***L= Lecture, *T= Tutorial, *P= Practical, *O= Others, F2F*= Face to Face, NF2F*= Non Face to Face**

16 . **Identify Special Requirement to Deliver the Course (e.g., software, nursery, computer lab, simulation room):**
Computer lab

17 . **Main References:**
Robert W. Sebesta. Concepts of Programming Languages, 10nd ed, Addison Wesley, 2012.

18 . **Additional References:**
Allen B. Tucker. Programming Languages: Principles and Paradigms, McGraw-Hill Higher education, 2007.
Franklyn A. Turbak, David K. Gifford. Design Concepts in Programming Languages. The MIT Press, 2008.
Michael L. Scott. Programming Language Pragmatics , 3rd ed. Morgan Kaufmann, 2009.

**Note:**

| | |
|---|---|
| ▨ | Cells shaded light grey contain formulas / fixed values. Edit these formulas only if needed. |