

Bachelor of Science in Information Technology

**Course Syllabus**

1. **Course Overview**: Computing Fundamentals

CC1

3 Units

1. **Course Description**:

This course introduces the important principles of good programming and illustrates the methods for designing effective algorithms. Likewise the applications of knowledge representation techniques simulation, coding and testing of programs are introduced. Topics include fundamentals in logic formation, control structures, arrays, introduction to functions, and basics of programming using the Java programming language.

1. **Prerequisite**:

None

1. **University Vision**:

The University of the Cordilleras envisions itself as a premier university for the transformation of individuals into stalwart citizens with a noble sense of values in the service of humanity.

1. **University Mission**:

To develop and deliver relevant professional career curricula and developmental opportunities in a learner-centered environment.

1. **Program Education Objectives**:

|  |  |  |
| --- | --- | --- |
| **Code** | **Program Educational Objectives (PEO)** | **Vision & Mission** |
| IT1 | Use of ITE knowledge and skills to describe, conceptualize and solve problems based on available data. | / |
| IT2 | Ability to adopt to and respond to changes brought by modifications on requirements, available resources and circumstances. | / |
| IT3 | Competence in introducing new ideas through original and creative ways. | / |
| IT4 | Ability to work productively as a member of a team; manifestation of leadership skills. | / |
| IT5 | Acting in ways consistent with the norms of society, industries and individuals. | / |

1. **Student Outcomes**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Code** | **Student Outcomes (SO)** | **PEO** | | | | |
|  |  |  |  |  |
| SO1 | Apply knowledge of computing, science, and mathematics appropriate to the discipline. | / |  |  |  |  |
| SO2 | Understanding best practices and standards and their applications. | / |  |  |  |  |
| SO3 | Analyze complex problems, and identify and define the computing requirements appropriate to its solution. | / | / |  |  |  |
| SO4 | Identify and analyze user needs and take them into account in the selection, creation, evaluation, and administration of computer-based systems. | / | / | / |  |  |
| SO5 | Design, implement, and evaluate computer-based systems, processes, components, or programs to meet desired needs and requirements under various constraints. | / | / |  |  |  |
| SO6 | Integrate IT-based solutions into the user environment effectively. | / | / |  |  |  |
| SO7 | Apply knowledge through the use of current techniques, skills, tools, and practices necessary for the IT profession. | / | / |  |  |  |
| SO8 | Function effectively as a member or leader of a development team recognizing the different roles within a team to accomplish a common goal. |  |  | / |  |  |
| SO9 | Assist in the creation of an effective IT project plan. | / | / | / |  |  |
| SO10 | Communicate effectively with the computing community and with society at large about complex computing activities through logical writing, presentations, and clear instructions. | / | / | / |  |  |
| SO11 | Analyze the logical and global impact of computing information technology on individuals, organizations, and society. | / | / | / | / |  |
| SO12 | Understand professional, ethical, security and social issues, and responsibilities in the utilization of information technology. | / |  |  | / |  |
| SO13 | Recognize the need for and engage in planning self-learning and improving performance as a foundation for continuing professional development. |  |  | / |  | / |

1. **Course Outcomes**:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Code** | **Course Outcomes (CO)** | **SO** | | | | | | | | | | | | | **PEO** | | | | |
| **SO1** | **SO2** | **SO3** | **SO4** | **SO5** | **SO6** | **SO7** | **SO8** | **SO9** | **SO10** | **SO11** | **SO12** | **SO13** | **IT1** | **IT2** | **IT3** | **IT4** | **IT5** |
| CO1 | Define and identify technical terms used in programming. | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I |
| CO2 | Compare and contrast operators used in programming. | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I | I |
| CO3 | Differentiate control structures used in programming . | E | E | E | E | E | E | E | E | E | E | E | E | E | E | E | E | E | E |
| CO4 | Create and test programs solving basic mathematical problems. | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D | D |

**Legend**: **I** – Introduction **E** – Enabling **D** – Demonstration

1. **Grading System**:
2. Midterm Score = 33% Midterm Class Standing + 33% Midterm Laboratory Scores + 34% Midterm Exam
3. Final Score = 50% Midterm Score + 50% Tentative Final Score

* Tentative Final Score = 33% Final Class Standing + 33% Final Laboratory Scores + 34% Final Exam

**Note:** 1. CS is composed of the course requirements and the required online courses.

2. Scores are transmuted to an equivalent grade where a score of 50% is needed to get a passing grade of 75.

1. **Course Content**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Topics** | **Time Frame** | **Teaching Strategies** | **Assessment** |
| **1. Programming Concepts**   * Terminolgies * Program Flowcharting * Basic programming hints * Compiling and executing a java source code * Type of errors | 8 hrs, 20 mins | Online Lecture  Coding Demonstrations  Self-Paced  Reading Assignments | Module Assignments |
| **2. Java Programming Structure**   * Statements and Blocks * Identifiers * Constants and Variables * Primitive Data Types | 8 Hrs, 20 mins. | Online Lecture  Coding Demonstrations  Self-Paced  Reading Assignments | Module Assignements |
| **3. Basic Operators**   * Assignment * Arithmetic * Relational * Logical * Precedence and Associativity of Operators | 15 hrs. | Online Lecture  Coding Demonstrations  Self-Paced  Reading Assignments | Module Assignements |
| **4. Sequential Control Structure**   * Sequential Flow * Sequential Structure | 8 hrs, 20 mins. | Online Lecture  Coding Demonstrations  Self-Paced  Reading Assignments | Module Assignements |
| **5. Decision Control Structure**   * If * If-else * Switch-case * Nested vs. Cascading if/else * Sequential Structure | 8 hrs, 20 mins. | Online Lecture  Coding Demonstrations  Self-Paced  Reading Assignments | Module Assignements |
| **6. Repetition Control Structures**   * While loop * For loop * Do-while loop * Counters and Accumulators | 6 hrs, 40 mins. | Online Lecture  Coding Demonstrations  Self-Paced  Reading Assignments | Module Assignements |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **MIDTERM EXAMINATION (1 Hr, 30 mins.)** | **Topic Learning Outcomes (TLO)** | **CO1** | **CO2** | **CO3** | **CO4** | | 1. To identify symbols used in Java programming | / | / | / | / | | 1. To differentiate the different control structures | / | / | / | / | | 1. To debug Java statements | / | / | / | / | | 1. To use correct conditional statements | / | / | / | / | |  | 1. To simulate the correct output | / | / | / | / | | | | |
| **7. Java Arrays**   * Terminologies * 1 and 2 Dimensional Arrays | 15 Hours | Online Lecture  Coding Demonstrations  Self-Paced  Reading Assignments | Module Assignements |
| **8. Java Methods**   * Terminologies * Creating Methods | 5 Hrs, 40 mins. | Online Lecture  Coding Demonstrations  Self-Paced  Reading Assignments | Module Assignments |
| **8. Exception Handling**   * Try-Catch | 1 Hr | Research | Research Output |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **FINAL EXAMINATION (1 Hour, 30 mins.)** | **Topic Learning Outcomes (TLO)** | **CO1** | **CO2** | **CO3** | **CO4** | | 1. To debug and existing Java array source code | / | / | / | / | | 1. To create a syntactically and semantically correct Array source codes | / | / | / | / | | 1. To create methods in solving mathematical problems | / | / | / | / | |  |  |  |  |  | | | | |

1. **References**:
2. Farrel, Programming Logic and Design – Comprehensive, 4th edition, Thomson Course Technology, c2009
3. Farrel, Java Programming, 5th edition, Cengage Learning, c2010
4. Malik, DS Java Programming: From Problem Analysis to Program Design, 6th edition, Boston Course Technology c2010
5. Smith, Java Programs to accompany Programming Logic and Design, 2nd edition, Boston Course Technology c2010
6. Horstmann, Cornell, Core Java; Microsystems, Inc, California, c2009

Prepared by: Recommending Approval: Approved by:

**Felipe A. Pati, Jr. Natividad B. Concepcion Jeffrey S. Ingosan**

Faculty Program Chair, CS Academic Dean