

**COLLEGE OF INFORMATION TECHNOLOGY**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**COURSE SYLLABUS/SPECIFICATION**

|  |  |  |
| --- | --- | --- |
| **CODE& TITLE:** | **ITCS 201 – Object Oriented Programming I** | |
| **WEIGHT:** | **(2 - 2 - 3)** |  |
| **PREREQUISITE:** | **ITCS 122** |  |
| **DESCRIPTION:** | This course emphasizes on object oriented programming techniques using Java. It covers the implementation of object oriented concepts, such as: classes, objects, inheritance and polymorphism. | |
| **OBJECTIVES:** | 1. To develop an understanding of the principles underpinning object oriented programming.  2. To designate the important features of an object oriented programming language.  3. To understand how to design and implement object oriented concepts and software. | |
| **SEMESTER:** |  | **ACADEMIC YEAR:** |
| **INSTRUCTOR:** | | |
| **OFFICE TEL.:** | | |
| **EMAIL:** | | |

**INTENDED LEARNING OUTCOMES (ILOS)**

Upon successful completion of the course, students should be able to:

|  |  |
| --- | --- |
| **A. Knowledge and Understanding** | |
| **A1** | Concepts and Theories: Demonstrate detailed knowledge and understanding of the concepts involved in Java programming for implementing object oriented software. |
| **A2** | Contemporary Trends, Problems and Research: NA |
| **A3** | Professional Responsibility: NA |

|  |  |
| --- | --- |
| **B. Subject-Specific Skills** | |
| **B1** | Problem Solving: Solve defined and some undefined problems by writing Java programming code. |
| **B2** | Modeling and Design: Design the prototype for solving different kinds of real world  problems, by clearly stating the concepts involved; such as Constructors, Abstract class, Inheritance, Polymorphism, Interface and their respective set of Properties and Methods using algorithms and UML models. |
| **B3** | Application of Methods and Tools: Apply Java programming language constructs and tools to write, run, trace, and debug object oriented programs. |

|  |  |
| --- | --- |
| **C. Thinking Skills** | |
| **C1** | Analytic: Evaluate and find the best Java object oriented concepts needed to develop efficient and effective programs. |
| **C2** | Synthetic: Integrate different application objects and object oriented concepts into complete computer applications needed to solve real world problems. |
| **C3** | Creative: NA |

|  |  |
| --- | --- |
| **D. General and Transferable Skills (Other Skills Relevant to Employability and Personal**  **Development)** | |
| **D1** | Communication: Demonstrate the ability to present Java programs clearly in a well structured manner. |
| **D2** | Teamwork and Leadership: NA |
| **D3** | Organizational and Developmental Skills: Demonstrate ability to organize ideas and  effectively allocate time in given assignments. |
| **D4** | Ethical and Social Responsibility: NA |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Structure (Outline)** | | | | | | | |
| **Week** | **Hours**  **c.** | | | **ILOs** | **Unit/Module or Topic Title** | **Teaching Method** | **Assessment**  **Method** |
|  | **Lecture** | | **Lab** |  |  |
| 1 | 2 |  | 2 | A1 | Introduction to object oriented  programming language. | Lecture |  |
| 2 | 2 |  | 2 | A1, B2, B3 | Introduction to Classes,  Objects, Methods and Instance Variables, Default, Public, and Private Access Modifiers, UML Notations. | Lectures/ lab demonstration | In-Lab  Exercises |
| 3 | 2 |  | 2 | B3 | Declaring a Class and  Instantiating an Object, Setter, Getter, and Operational Methods. | Lecture/ lab demonstration | In-Lab  Exercises |
| 4 | 2 |  | 2 | A1, B3, D1, D3 | Constructor, Default, No- Argument and Arguments Constructors, Initializing Objects with Constructors. | Lecture/ Lab  demonstration  / In-Lab Supervised Work | Assignment 1 |
| 5 | 2 |  | 2 | A1, B1, B3, C1 | Overloading Methods, this Keyword, Encapsulation and Data Hiding. | Lecture/ Lab  demonstration  / In-Lab Supervised Work | In-lab  Exercises |
| 6 | 2 |  | 2 | A1, B1, B3, C1 | Static Variables, Static Methods, Static Class Members, Static Import. | Lecture/ Lab  demonstration  / In-Lab Supervised Work | In-lab  Exercises |
| 7 | 2 |  | 2 | A1, B1, B2, B3, C1 | Enumerations, Garbage  Collection and Method finalize, Final Instance Variables, Creating Packages, Package Access, and UML Package Notations. | Lecture/ Lab demonstration  / In-Lab Supervised Work | In-lab  Exercises |
| 8 | 2 |  | 2 | A1, B1, B2, B3, C1 | Association, Aggregation Relationships and UML Notation. | Lecture/ In-  Lab Supervised Work | Lab Test 1 |
| 9 | 2 |  | 2 | A1, B1, B2, B3, C1, C2, D1, D3 | Inheritance and UML Notation. | Lecture/ Lab  demonstration  / In-Lab Supervised Work | In-lab  Exercises |
| 10 | 2 |  | 2 | A1, B1, B3 | Protected Access Modifier, | Lecture/ Lab | In-Lab |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | Software Engineering with  Inheritance, Object Class and super Keyword. | demonstration  / In-Lab Supervised Work | Exercises |
| 11 | 2 | 2 | A1, B1, B3, C1 | Polymorphism and Overriding  Methods. | Lecture/ Lab  demonstration  / In-Lab Supervised Work | In-Lab  Exercises |
| 12 | 2 | 2 | A1, B1, B2, B3, C1, C2, D1, D3 | Static and Dynamic Binding. | Lecture/ Lab  demonstration  / In-Lab Supervised Work | In-Lab  Exercises |
| 13 | 2 | 2 | A1, B1, B2, B3, C1, C2 | Abstract Class, Abstract Method and UML Abstract Notation. | Lecture/ In-  Lab Supervised Work | Lab Test 2 |
| 14 | 2 | 2 | B1, B3 | Final Methods and Classes. | Lectures/ lab  Demonstratio n | In-Lab  Exercises |
| 15 | 2 | 2 | A1, B1, B2, B3, C1, C2, D1, D3 | Interfaces, Common Interfaces of the Java API and UML Interfaces Notation. | lab  Demonstratio n/ In-Lab Supervised Work | Assignment 2 |
| 16 | 2 | - | A1, B1, B2,  C1 | All Topics |  | Final Exam |

**TEACHING MATERIALS:**

|  |  |
| --- | --- |
| **TEXTBOOK(S):** | Deitel T. R. Nieto. (2015) *Java How to Program*, 10th Edition, Prentice Hall. |
| **HANDOUT(S):** | **-** Internal handouts (Hardcopies) prepared by course instructors.  - PowerPoint slides available on Moodle i.e. <http://www.ahlia.edu.bh/moodle> |
| **REFERENCE(S):** | 1. Gaddis T. and Muganda G. (2015) *Starting Out with Java, From Control*  *Structures through Data Structures*, 3rd Edition, Pearson Education.  2. Liang Y. D. (2013) *Introduction to Java Programming, Comprehensive Version plus My Programming Lab with Pearson eText - Access Card*, 9th Edition Prentice Hall.  3. Barnes D. J. and Kölling M. (2012) *Objects First with Java: A Practical*  *Introduction Using BlueJ,* 5th Edition, Prentice Hall.  4. Litvin M. and Litvin G. (2015) *Java Methods: Object-Oriented*  *Programming and Data Structures*, 3rd AP edition, Skylight Publishing. |

**WEBSITES:** [http://www.planet-source-code.com](http://www.planet-source-code.com/) [http://www.java-examples.com](http://www.java-examples.com/)

**ASSESSMENTS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of**  **Assessment** | **Description** | **ILOs** | **Weighting** |
| Assignments | Each group of students is required to  prepare two assignments. Assignments include both theoretical and practical questions to develop an object oriented program for solving a problem. | A1, B1, B2, B3, C1, C2, D1, D3 | 20% |
| Lab Tests | Two practical tests will be for two hours  and worth 20% each, The tests used to assess students in the implementation of object-oriented programs using Java. | B1, B3, C1, C2 | 40% |
| Final Exam | Final exam will be for two hours and  including all types of question: problem solving, MCQs and T/F, short answers, programming. | A1, B1, B2, C1 | 40% |
| In-Lab Exercises | The students will practice through a number  of exercises to design and implement the object oriented programming concepts. | B1, B2, B3 | Formative |
| **Overall** |  |  | **100%** |

***Ahlia University values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see*** [***www.ahlia.edu.bh/integrity***](http://www.ahlia.edu.bh/integrity) ***for more information).***