



Course Specifications

Course Title:	Object Oriented Programming
Course Code:	501323-3
Program:	Bachelor in Computer Science
Department:	Department of Computer Science
College:	College of Computers and Information Technology
Institution:	Taif University

Table of Contents

A. Course Identification	3	
6. Mode of Instruction (mark all that apply)		3
B. Course Objectives and Learning Outcomes	3	
1. Course Description		3
2. Course Main Objective		3
3. Course Learning Outcomes		3
C. Course Content	4	
D. Teaching and Assessment	4	
1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods		4
2. Assessment Tasks for Students		4
E. Student Academic Counseling and Support	5	
F. Learning Resources and Facilities	5	
1. Learning Resources		5
2. Facilities Required		5
G. Course Quality Evaluation	5	
H. Specification Approval Data	6	



A. Course Identification

1. Credit hours: 3
2. Course type
a. University <input type="checkbox"/> College <input checked="" type="checkbox"/> Department <input type="checkbox"/> Others <input type="checkbox"/>
b. Required <input checked="" type="checkbox"/> Elective <input type="checkbox"/>
3. Level/year at which this course is offered: 8/3
4. Pre-requisites for this course (if any): 501222-3 (Programming 2)
5. Co-requisites for this course (if any): NON

6. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	5	62.5%
2	Blended	0	0
3	E-learning	0	0
4	Distance learning	0	0
5	Other (Lab)	3	37.5%

7. Contact Hours (based on academic semester)

No	Activity	Contact Hours
1	Lecture	50
2	Laboratory/Studio	30
3	Tutorial	0
4	Others (specify)	0
	Total	80

B. Course Objectives and Learning Outcomes

1. Course Description

This course provide Object-oriented programming paradigm, focusing on core concepts like Classes, Objects, Encapsulation, Inheritance, Polymorphism, etc., by using an object-oriented language. It introduces also UML Class Modeling tools to design and implement object-oriented applications.

2. Course Main Objective

Students at the end of this course are able :

- Justify the philosophy of object-oriented design and the concepts of class, object, encapsulation, abstraction, inheritance, and polymorphism.
- Describe how the class mechanism supports encapsulation and information hiding.
- Design, implement, test, and debug simple programs in an object-oriented programming language.



3. Course Learning Outcomes

CLOs		Aligned PLOs
1	Knowledge and Understanding:	
1.1	Understand Object Oriented Programming (OOP) paradigm	K1
1.2	Understand UML Class Presentation for OOP applications	K1
2	Skills:	
2.1	Use OOP features and tools to design applications	S2
2.2	Implement object oriented applications	S2
3	Values:	

C. Course Content

No	List of Topics	Contact Hours
1	Course overview, policies, Basic concepts of OOP	5
2	Introduction to Java Programming	5
3	OOP Fundamentals: Classes, Objects, states , Methods , Encapsulation and UML Class Presentation	5
4	OOP Features: Driver class, Access modifier, Constructors, Instantiation, Message Passing, etc.	5
5	Composition Concept : Reuse of code by modeling UML <i>has-a</i> composition's relation	5
6	Classes and Objects Deeper Look - Part 1 : Static Class Members, Static import, Math static methods examples	5
7	Classes and Objects Deeper Look - Part 2 : Access modifier & Packages and Garbage	5
8	Inheritance Concept : Reuse of code by modeling UML <i>is-a</i> extending's relation	5
9	Inheritance Deeper Look: Method overriding, final Methods and final Classes, Abstract Classes and Methods, Polymorphism	7
12	Exception Handling	3
Total		50

D. Teaching and Assessment

1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.0	Knowledge and Understanding		
1.1	Understand Object Oriented Programming paradigm	Lectures Labs Project	Direct Assessment Tool Quizzes / Homework/Project/ Exams Indirect Assessment Tool Course Exit Survey



Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.2	Understand UML Class Presentation for OOP applications	Lectures Labs Project	Direct Assessment Tool Quizzes / Homework/Project/ Exams Indirect Assessment Tool Course Exit Survey
2.0	Skills		
2.1	Use OOP features and tools to design applications	Lectures Labs Project	Direct Assessment Tool Quizzes / Homework/Project/ Exams Indirect Assessment Tool Course Exit Survey
2.2	Implement object-oriented applications	Lectures Labs Project	Direct Assessment Tool Quizzes / Homework/Project/ Exams Indirect Assessment Tool Course Exit Survey
3.0	Values		

2. Assessment Tasks for Students

#	Assessment task*	Week Due	Percentage of Total Assessment Score
1	HomeWorks /Student Participation-Attendance	Every Week	5%
2	Project	Week 2 □ 10	15%
3	Quizzes	Week 2, 4, 6, 8	10%
4	Final Labs Exam	Week 10	10%
5	Mid-Term	Week 5	20%
6	Final Examination	Week 12	40%

*Assessment task (i.e., written test, oral test, oral presentation, group project, essay, etc.)

E. Student Academic Counseling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice :

Academic advising and counseling of students is an important component of teaching; student academic advising is a mandatory requirement of College of Computers and Information Technology (CCIT). Appropriate student advising provides support needed for the student during times of difficulty. In addition, it helps the student to build a close relationship with his/her advisor and to provide student motivation and involvement with the institution.

In addition, since faculty are usually the first to recognize that a student is having difficulty, faculty members play a key role in developing solutions for the students or referring them to appropriate services. Faculty members also participate in the formal student-mentoring program.



Additional counseling is provided by course directors, who provide students with academic reinforcement and assistance and refer “at risk” students to the Vice Dean for Academic Affairs and the Vice Dean for female section.

F. Learning Resources and Facilities

1. Learning Resources

Required Textbooks	Java How to Program, by Deitel, P., 9th Edition Prentice Hall. ISBN-13: 9780132575669
Essential References Materials	<ul style="list-style-type: none"> • Introduction to Java Programming, Comprehensive Version by Y. Daniel Liang Publisher: Prentice Hall; 10th edition (January 6, 2014). ISBN-10: 0133761312 ISBN-13: 978-0133761313 • Java: An Introduction to Problem Solving and Programming, (6th Edition) by Walter Savitch • Building Java Programs (3rd Edition) by S. Reges & M. Stepp Publisher, Pearson 2013. ISBN-13: 978-0133360905
Electronic Materials	<ul style="list-style-type: none"> • Introduction to Programming using Java by David Eck. (2011). Online edition. • Introduction to Programming in Java by R. Sedgewick & K. Wayne. Online edition
Other Learning Materials	<ul style="list-style-type: none"> • http://www.eclipse.org/downloads/ • https://netbeans.org/downloads • ObjectAid UML Explorer (http://www.objectaid.com/update/current)

2. Facilities Required

Item	Resources
Accommodation (Classrooms, laboratories, demonstration rooms/labs, etc.)	<ul style="list-style-type: none"> • A Lecture room appropriate for maximum 25 students with a personal computer, a data show and a smart board. • A Lab room appropriate for maximum 15 students with a personal computer, a data show and a smart board.
Technology Resources (AV, data show, Smart Board, software, etc.)	<ul style="list-style-type: none"> • Lab materials and required software
Other Resources (Specify, e.g. if specific laboratory equipment is required, list requirements or attach a list)	

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of Teaching	Students	Students' surveys and Student's course evaluation
Improvement of Teaching	Course Coordinator	deficiencies based on the student Evaluation, faculty input, course file, and program assessment
Verifying Standards of Student Achievement	Curriculum Committee	<ul style="list-style-type: none"> Review CAF (Course assessment file) Alumni surveys. Periodic exchange and remarking of tests or a sample of assignments with staff at another

Evaluation areas (e.g., Effectiveness of teaching and assessment, Extent of achievement of course learning outcomes, Quality of learning resources, etc.)

Evaluators (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify))

Assessment Methods (Direct, Indirect)

H. Specification Approval Data

Council / Committee	CS council
Reference No.	Meeting #12
Date	23-10-1443

