



Course Specifications

Course Title:	Computer Programming 2
Course Code:	CP41
Program:	Diploma in Programming and Computer Sciences
Department:	Technology department
College:	Applied College
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	4
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	5
E. Student Academic Counseling and Support	5
F. Learning Resources and Facilities	6
1. Learning Resources	6
2. Facilities Required.....	6
G. Course Quality Evaluation	7
H. Specification Approval Data	7

A. Course Identification

1. Credit hours:	4
2. Course type	
a.	University <input type="checkbox"/> College <input type="checkbox"/> Department <input checked="" 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:	Second Year / Fifth Level
4. Pre-requisites for this course (if any):	CP31-Computer Programming 1
5. Co-requisites for this course (if any):	None

6. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	6	100%
2	Blended		
3	E-learning		
4	Distance learning		
5	Other (Lab)		

7. Contact Hours (based on academic semester)

No	Activity	Contact Hours
1	Lecture	30
2	Laboratory/Studio	30
3	Tutorial	
4	Others (specify)	
	Total	60

B. Course Objectives and Learning Outcomes

<p>1. Course Description</p> <p>In this course on computer programming students are taught an introduction to object-oriented concepts and techniques such as classes, interfaces, encapsulation, inheritance, and polymorphism using object-oriented languages. Introduce UML Class Modeling to describe an object-oriented programming approach.</p>
<p>2. Course Main Objective</p> <p>Students at the end of this course are able to :</p> <ul style="list-style-type: none"> Justify the philosophy of object-oriented design and the concepts of class, object, encapsulation, abstraction, inheritance, and polymorphism. Design, implement, test, and debug simple programs in an object-oriented programming language. Describe how the class mechanism supports encapsulation and information hiding.

3. Course Learning Outcomes

CLOs		Aligned PLOs
1	Knowledge and Understanding	
1.1	Write the basic concepts of Class and object, methods, overloading, constructors, inheritance and polymorphism, abstract class, exception handling, and interfaces with inner classes in the programming language.	K2
1.2	Define inheritance, polymorphism, abstract class, and exception handling , interfaces with inner classes in the programming language.	K2
2	Skills :	
2.1	Design programs and classes using the concepts of object oriented programming like class and object, inheritance, polymorphism, abstract class and interfaces.	S1
2.2	Implement programs that use concepts of object oriented programming	S1
3	Values:	
3.1	Work individual or in groups to solve a common problem.	V1

C. Course Content

No	List of Topics (Lab an Theory)	Contact Hours
1	Introduction: Introduction to Java OO programming.	6
2	Objects and Classes:Part1	6
3	Objects and Classes:Part2	6
4	Thinking in Objects	12
5	Inheritance and Polymorphism: part1	12
6	Inheritance and Polymorphism: par2	12
7	Abstract Classes	6
8	Interfaces	6
9	Exception Handling	6
10	Review	6
Total		60

D. Teaching and Assessment

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

Code	Course Learning Outcomes	TeachingStrategies	AssessmentMethods
1.0	Knowledge and Understanding		
1.1	Write the basic concepts of Class and object, methods, overloading, constructors, inheritance and polymorphism, abstract class, exception handling, and interfaces	Lectures	Quizzes /Exams Homework /Evaluation

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
	with inner classes in the programming language.		
1.2	Define inheritance, polymorphism, abstract class, and exception handling , interfaces with inner classes in the programming language.	Lectures	Quizzes /Exams Homework /Evaluation
2.0	Skills		
2.1	Design programs and classes using the concepts of object oriented programming like class and object, inheritance, polymorphism, abstract class and interfaces.	Lectures Labs Project	Direct Quizzes/ Homework/ Project/ Assignments/ Lab Exam /Exams Indirect Course Exit Survey
2.2	Implement programs that use concepts of object oriented programming.	Lectures Labs Project	Direct Assessment Tool Lab Exams/ Exams / Project Indirect Assessment Tool Course Exit Survey
3.0	Values		
3.1	Work individual or in groups to solve a common problem.	Lectures Labs Project	Direct Assessment Tool Lab/ Exams / Project/ Assignments/ Lab Exam/ Indirect Assessment Tool Course Exit Survey

2. Assessment Tasks for Students

#	Assessment task*	Week Due	Percentage of Total Assessment Score
1	Mid-Term	6	20%
2	Quizzes	4	5%
3	Home Works/ Attendance / Evaluation/project	Every Week	10 %
4	Final Labs Exam	11	15%
5	Final Examination	12	50%

*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 :

- ✓ Providing a guide for each group of students, and distributing student lists electronically to faculty members.
- ✓ There is an academic advising guide that defines the role of the faculty member in the academic advising process.
- ✓ The program supervisor is available throughout the year to answer student inquiries.
- ✓ Availability of full information about the program and its members and ways to communicate with them.
- ✓ Use the Learning Management System (Black Board) to communicate with students
- ❖ Student Handbook, Deanship of Student Affairs.
https://www.tu.edu.sa/Attachments/893d1c33-3156-44d7-b4b8-e203d4cca737_.pdf
- ❖ Student Handbook at Taif University.
https://www.tu.edu.sa/Attachments/41dc8a24-22b7-4ae1-9b31-3608de8bcf8b_.pdf

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, (6 ThEdition) 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"> ❖ Classroom with 50 chairs ❖ Lab with 25 chairs
Technology Resources (AV, data show, Smart Board, software, etc.)	<ul style="list-style-type: none"> ❖ Availability of a Data Show ❖ Provides a smart board. ❖ Provide a whiteboard and colored whiteboard pens.
Other Resources (Specify, e.g. if specific laboratory equipment is required, list requirements or attach a list)	❖ Null

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of teaching	<ul style="list-style-type: none"> • Students • Faculty members • Coordinator • Program Leaders 	<ul style="list-style-type: none"> • Course exit survey • Feedback from Faculty members • Feedback from Course Coordinator • Feedback from Quality Committees
Effectiveness of assessment	<ul style="list-style-type: none"> • Faculty members • Coordinator • Program Leaders 	<ul style="list-style-type: none"> • Feedback from Faculty members • Feedback from Course Coordinator • Feedback from Program Leader
Extent of course achievement	<ul style="list-style-type: none"> • Students • Coordinator • Faculty members 	<ul style="list-style-type: none"> • Course exit survey • Curriculum Committees • Feedback from Course Coordinator • Feedback from Program Leader
Extent of course learning outcomes	<ul style="list-style-type: none"> • Faculty members • Coordinator • Program Leaders • Quality Committees 	<ul style="list-style-type: none"> • Course exit survey • Curriculum Committees • Feedback from Course Coordinator • Feedback from Program Leader • Feedback from Quality Committees
Quality of learning resources	<ul style="list-style-type: none"> • Students • Faculty members • Coordinator 	<ul style="list-style-type: none"> • Course exit survey • Course exit survey • Curriculum Committees • Feedback from Course Coordinator • Feedback from Program Leader

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	
Reference No.	
Date	