Ministry of Higher Education and Scientific Research

Supervision and Scientific Evaluation Body

Quality Assurance and Academic Accreditation Office

جامعة شط العرب
كلية العلوم
قسم علوم الحاسوب

Course Description Sample

Subject: Object oriented programming I

This course description provides a concise summary of the main features of the course and the learning outcomes expected of the student, demonstrating whether the student has made the most of the learning opportunities available. It must be linked to the programme description.

1. Educational Institution	Shatt Al Arab University
2. Department / Center	Computer Science
3. Course Title /Code	Object oriented programming I
4. Lecturer Name	**Assistant Lecturer Hussein Mazin Mohammed**
5. Type of Teaching	Theory – Lab – Tutorial
6. Academic Year /Term	2024/2025
7. Total No. of Teaching Hours	98
8. Date f Preparing this Course Description	5/8/2025

9. **Course Objectives**

By the end of this course, the student will be able to:

1. Explain the basic concepts of simple to intermediate-level website development.
2. Use standard web page languages (such as HTML and CSS) to create a simple website.
3. Design interactive web pages using appropriate web editors.
4. Identify the basic programming languages used in web development and their features.
5. Apply object-oriented programming concepts using Java, including:

Design and implement objects and classes, apply the concepts of construction and destructuring, employ the concepts of encapsulation and inheritance, and analyze and use the concept of polymorphism in programming.

10. **Course Output, Methodology and Evaluation**

(A) **Cognitive Objectives**

By the end of this course, the student will be able to:

1. Explain the basic concepts of simple to intermediate-level website development.
2. Use standard web page languages (such as HTML and CSS) to create a simple website.
3. Design interactive web pages using appropriate web editors.
4. Identify the basic programming languages used in web development and their features.
5. Apply object-oriented programming concepts using Java, including:

Design and implement objects and classes, apply the concepts of construction and destructuring, employ the concepts of encapsulation and inheritance, and analyze and use the concept of polymorphism in programming.

(B) **Skill Objectives Related to the Program**:

- Remembering
  - Identify the basic programming language used in web development and its characteristics.
- Understanding
  - Explain the fundamental concepts of developing simple to medium-level websites.
- Applying
  - Use standard web page languages (HTML and CSS) to create a simple website.
  - Apply object-oriented programming (OOP) concepts using Java, including:
    - Designing and implementing objects and classes
    - Applying constructors and destructors
    - Using encapsulation
    - Implementing inheritance
    - Analyzing and applying polymorphism
- Analyzing
  - Analyze and apply the concept of polymorphism to distinguish between different object behaviors.
  - Differentiate between webpage components (structure, design, interactivity) and their relation to user experience.
- Creating
  - Design interactive web pages using appropriate web editors.
  - Develop programs or websites that integrate multiple programming concepts (OOP + user interface + simple databases).
- Evaluating
  - Evaluate the quality of web pages in terms of design, usability, and compliance with web standards.

- Judge the efficiency of code in terms of readability, scalability, and maintainability.

Methods of Teaching and Learning

1-Lecturers.
2-Class discussion.
3-Lab Experiments.
4-Researchs.
5-Homework.

Methods of Evaluation

Number	calendar element	degree
1-Examinations.		
2-Lab Experiments.		
3-Quizz.		
4-Oral Exam.		
5-Researchs.		

(C) **Sentimental and Value Objectives**

Affective Objectives
• Develop **intrinsic motivation** toward learning programming and web development as a tool for creativity and problem-solving. • Demonstrate **interest** in improving programming skills and keeping up with technological advancements. • Foster **self-confidence** in the ability to design and implement programming projects. • Promote **teamwork and collaboration** when working on group programming projects. • Show **respect for peers' opinions** when discussing programming solutions and exchanging knowledge.
Value-based Objectives • Commit to **academic integrity** when using or reusing code (avoiding plagiarism). • Enhance the value of **creativity and innovation** in developing new software solutions. • Emphasize **responsibility** in designing secure and reliable websites for users. • Uphold **ethical and professional standards** in software development. • Recognize the importance of **programming technologies** in serving society and contributing to education and the economy.

Methods of Teaching and Learning

Teaching and Learning Methods
1. Theoretical Lectures - Introducing basic concepts and modern technologies in smart applications. - A presentation supported by oral explanation. - Using real-world examples.

- Explaining the latest trends in artificial intelligence, the Internet of Things, and cloud applications.

2. Hands-on Learning
- Training students to design and implement advanced smart applications.
- Coding Labs to build mini-projects.
- Practical applications using languages and frameworks such as Python, Tensor Flow, Flutter, or React Native.

3. Project-Based Learning
- Enhancing applied and problem-solving skills.
- Assigning students to build an integrated smart application (such as one that recognizes images or relies on data analysis).
- Linking projects to real-world problems from the market or society.

4. Collaborative Learning
- Enhancing teamwork and communication skills.
- Dividing students into teams to complete joint tasks or projects.
- Organizing group discussions to solve programming problems.

5. Problem-Based Learning
- Sharpening critical thinking and technical challenge-solving skills.
- Presenting open-ended scenarios or problems to students.
- Asking them to propose innovative programming solutions using smart technologies.

6. Presentations & Discussions
- Improving presentation and persuasion skills.
- Students give presentations on advanced smart technologies or applications.
- Organizing discussion sessions to compare ideas and solutions.

7. Simulation-Based Learning
- Experiencing realistic work environments without real risks.
- Using simulation tools for IoT or AI systems. Test the performance of applications in a virtual environment before deploying them.

8. Blended Learning

- Combining traditional learning with e-learning.
- In-class lectures + interactive online content.

Using educational platforms such as Moodle or Google Classroom to distribute materials and track progress.

Methods of Evaluation

Module Evaluation تقييم المادة الدراسية					
		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	Quizzes	2	10% (10)	5, 10	LO #1, 2, 10 and 11
	Assignments	2	10% (10)	2, 12	LO # 3, 4, 6 and 7
	Projects / Lab.	1	10% (10)	Continuous	
	Report	1	10% (10)	13	LO # 5, 8 and 10
Summative assessment	Midterm Exam	2 hr	10% (10)	7	LO # 1-7
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

D) **General and Qualitative Skills (other skills related to the ability of employment and personal development)**

D - General and Transferable Skills (other skills related to employability and personal development).

The developed curriculum focuses not only on academic knowledge but also on developing general skills that enhance students' employability and personal development. These skills include:

1. Analytical and Problem-Solving Skills
• Analyzing the behavior of swarm systems (such as ant colonies or bird swarms) and deriving mathematical rules from them.
• Designing optimization algorithms such as PSO or ACO to solve real-world problems (such as logistics delivery or energy management).

Benefits in the Job Market:

• The ability to address complex problems in fields such as artificial intelligence, robotics, and automation.

2. Creative Thinking and Innovation
• Encouraging students to propose new applications for swarm intelligence or fuzzy logic (such as their use in smart games or digital health).
• Designing open-ended projects (such as "How would a swarm of firefighting robots operate in a forest?").

Benefits in the Job Market:
• Enhancing innovation capacity in sectors such as financial technology (Fintech) and smart cities.

3. Teamwork and Communication

• Group projects (such as a drone swarm simulation) that require task allocation and coordination.
• Presentations explaining practical results using simple, technical terms.
Benefits in the job market:
• Ability to work in multidisciplinary teams (developers, engineers, data analysts).

4. Programming and Technical Skills
• Using Python to implement swarm intelligence algorithms
• Familiarity with systems simulation tools (such as MATLAB)
Benefits in the job market:
• Increased job opportunities for positions such as "Machine Learning Engineer" or "Intelligent Systems Developer."

5. Project Management
• Dividing projects into phases (planning, implementation, evaluation) with set deadlines.
• Using tools such as Trello or GitHub to manage programming tasks.
Benefits in the job market:
• The ability to manage technical projects from start to finish.

6. Adapting to Changing Technology

• Discussing the latest research in swarm intelligence (such as combining it with deep learning).
• Weekly challenges to solve problems using emerging technologies.
Benefits in the job market:
• Staying abreast of rapid developments in fields such as artificial intelligence and robotics.

7. Quantitative and Mathematical Skills

• Applying mathematical optimization equations (such as velocity update equations in PSO).
• Analyzing simulation data using Simple statistics.
Its benefits in the job market:
• Useful in roles such as "data analyst" or "artificial intelligence researcher."

11. **Course Structure**

(In the table of course weekly outline)

Week	No of Hours	Required Learning Output	Title of Subject	Teaching Method	Evaluation
1			Programming style	• In-person lectures • Practical laboratory lectures • Reports • Tests	
2			Basic statements with looping and repetitions	• In-person lectures • Practical laboratory lectures • Reports • Tests	
3			One dimensional Arrays	• In-person lectures • Practical laboratory lectures • Reports • Tests	
4			Two dimensional Arrays	• In-person lectures • Practical laboratory lectures	

				• Reports • Tests	
5			Classes and methods	• In-person lectures • Practical laboratory lectures • Reports • Tests	
6			Classes and methods	• In-person lectures • Practical laboratory lectures • Reports • Tests	
7			Constructors, Variable types,	• In-person lectures • Practical laboratory lectures • Reports • Tests	
8			Overloading	• In-person lectures • Practical laboratory lectures • Reports • Tests	
9			UML diagrams	• In-person lectures • Practical laboratory lectures • Reports • Tests	
10			Overloading	• In-person lectures • Practical laboratory lectures • Reports • Tests	
11			UML Diagrams	• In-person lectures • Practical laboratory lectures • Reports • Tests	
12			Programming with Contracts: Preconditions, Postconditions, and Constants	• In-person lectures • Practical laboratory lectures • Reports • Tests	

13			Interface Design	• In-person lectures • Practical laboratory lectures • Reports • Tests	
14			Polymorphism	• In-person lectures • Practical laboratory lectures • Reports • Tests	
15			Encapsulation	• In-person lectures • Practical laboratory lectures • Reports • Tests	
16			Preparatory week before the final Exam	• In-person lectures • Practical laboratory lectures • Reports • Tests	

12.**Infrastructure**

a. Textbooks	Thomas Wu (2010). An Introduction to Object-Oriented Programming with Java. Fifth Edition. McGraw-Hill.
b. References	Herbert Schildt (2007). Java: The Complete Reference. Seventh Edition. McGraw-Hill.
c. Recommended books and periodicals (journals, reports, etc.)	Introduction to Object-Oriented Programming with Java.
d. Electronic references, internet websites, etc	The Collage E-Library

13. **The Plan of Improving the Course**

1. Add advanced skills 2. Link new skills to the overall knowledge objectives of the department

توقيع الاستاذ :

[illegible]

توقيع رئيس القسم :

[illegible]

توقيع رئيس القسم :

[illegible]

جامعة شط العرب
كلية العلوم
قسم علوم الحاسوب