

Course Specification Document

Title	Object Oriented Programming Techniques
--------------	--

Credits	3.5 ECTS
----------------	----------

Aims	This course aims to provide the student with knowledge of advanced object-oriented programming techniques and the fundamentals of building large programs, by focusing on object-oriented concepts, classes and objects, and mastering advanced programming tools in a pure object-oriented programming language.
-------------	---

Intended learning outcomes

On successful completion of this course, the student will be able to:

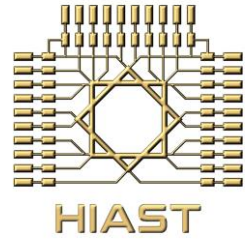
- Understand the concepts of objects and classes as essential components of large programs based on the encapsulation principle.
- Recognize the techniques for building classes in a pure object-oriented programming language.
- Understand the inheritance and its significance and types.
- Understand the polymorphism concept, its importance and the techniques used for its implementation and abstract classes and interfaces.
- Use an appropriate programming environment.
- Use the techniques of the object-oriented programming language to solve problems.
- Use graphical user interface (GUI) and user interaction programming to write a complete large software application with interactive graphical interfaces.

Syllabus

- **Introduction to advanced programming and large programs:** The importance of advanced programming in building large programs, small programs versus large programs, programming languages that support advanced programming, the importance of C# and its features.
- **Introduction to object-oriented programming:** The difference between procedural programming and object-oriented programming, the concept of abstraction, procedural abstraction and data abstraction, encapsulation and reuse, classes and objects.
- **Basics of Programming in C#:** Program structure, basic control structures, input and output instructions.
- **Definition of classes, methods and static members:** Definition of the class, class properties and methods, access modifiers, namespaces, method overloading, constructors, static members and methods.
- **Arrays, lists and text files:** Using practical programming techniques in object oriented programming languages.

Syrian Arab Republic

Higher Institute for Applied Sciences and Technology



- **Inheritance:** Base class and derived class, overriding methods in derived classes.
- **Polymorphism:** Importance and implications.
- **Interfaces and implementation:** Abstract classes, interfaces and implementation.
- **Exception handling:** Try-catch statement, throw statement, constructors, finalizers, exception handling.
- **Advanced programming topics:** Templates, generics and threads.
- **Graphical user interface (GUI):** Programming user interaction with form applications.