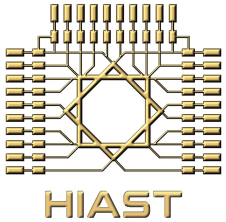
Syrian Arab Republic

Higher Institute for Applied Sciences and Technology



Course Specification Document

Title	Object Oriented Programming

Credits	5 ECTS

Aims	This course aims to introduce the student to the basics of object-oriented programming and to provide him with the necessary knowledge and skills to build computer programs based on this approach.

Intended learning outcomes

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

- Understand the meaning of object and class, and realize the benefit of applying an object-oriented approach to building software.
- Understand how to implement a class and build programs using objects.
- Understand the concepts of class composition and inheritance, and the difference between them.
- Understand the concepts of abstract classes, virtual functions, and polymorphism when building classes, and realize their advantages.
- Understand the concepts of reflections and exception handling.
- Use an appropriate programming environment for writing programs and the tools it provides to accelerate work, testing and experimentation.

Syllabus

- **Classes and objects:** Basic concepts, the concept of procedural and data abstraction, the structure, pillars of object oriented programming, class.
- **Constructors and destructors:** Function overloading, constructors (default constructor, constructors with argument, constructor implementation), destructors.
- **Methods:** Instance methods, static members and static methods.
- **Composite objects**: Object composition.
- **Inheritance:** Composition and inheritance, base class and derived class, overriding member functions in the derived class, constructors and destructors in derived classes, implicit conversion from a derived class to a base class.
- **Virtual functions and polymorphism:** Virtual functions, abstract and concrete classes, polymorphism, dynamic binding, virtual destructors.
- **Interfaces:** Interface definition, usage and applications.
- **Templates:** Function templates, class templates.
- **GUI applications:** Defining event-driven applications, Form class, using Form Designer.
- **Reflections**: Reflection concept, using reflection to analyze and modify objects at runtime.
- **Exception handling:** Exceptions and their types, dealing with exceptions.
- **Case study:** IOStream package.