

1. **Course code number and name:** 300CIS017, Object-Oriented Programming
2. **Credits and contact hours:** 3 credit hours, 5 hours per week
3. **Course Coordinator:** Luisa Rincón.
4. **Textbook:**

C++ programming: an object-oriented approach. Forouzan, B. A., & Gilberg, R. F, 2019.

Supplemental materials:

- Software Engineering with UML. Unhelkar, B. CRC Press. 2018
- Head First Object-Oriented Analysis and Design: A Brain Friendly Guide to OOA&D McLaughlin, B. D., Pollice, G., & West, D. 2007.
- C how to program, Sixth Edition; Deitel, Ed. Deitel, P., & Deitel, H. 2010.

5. Specific course information:

This course presents concepts of design and object-oriented programming (OO). The course uses the “Objects first” approach, which is the base of the OO paradigm. It also introduces notions of software engineering such UML class diagrams, design patterns, unitary testing and exception handling.

Prerequisites: 300CIP002, Foundations and Structures of Programming.

Type of course: Required.

6. Specific goals of the course:

Learning objectives:

- Identify the characteristics of object-oriented programming by studying techniques to represent and code classes, objects, methods, attributes with UML notations and the C++ language.
- Design and implement solutions to problems using the object-oriented paradigm by applying practices that favor abstraction, encapsulation, inheritance and polymorphism in C++ language.
- Recognize medium-scale software development techniques by studying exception management techniques, unit testing, software design principles and design patterns proposed in software engineering to build quality software products.

Relationship with student outcomes:

	Student Outcomes					
	1	2	3	4	5	6
Relevance	2	4			1	4

7. Topics of the course:

- Definition of classes.
- Core object concepts (Data abstraction, Encapsulation, Abstraction, Polymorphism)
- Access modifiers
- Unified Modelling Language (UML) and class diagrams
- Interaction between objects and message passing (association, dependency, composition and aggregation relationships).
- Use of data structures (sets, maps, iterators vectors, list).
- Abstraction techniques (abstract classes, multiple inheritance, interfaces).
- Inheritance
- Polymorphism
- Exception Handling
- Unitary testing
- GRASP principles
- OO Design patterns.

1. Course code number and name: 300CIS021, Emerging Technologies

2. Credits and contact hours: 2 credit hours, 3 hours per week.

3. Course Coordinator: Luisa Rincón.

4. Textbook:

This course does not have a guidebook

5. Specific course information:

This course familiarizes students with up-to-date industry trends to prepare them for their forthcoming job placement in industry. Therefore, the content changes over time to keep up to date with the industry trends.

Prerequisites: 300CIS017, Object-Oriented Programming.

Type of course: Required.

6. Specific goals of the course:

Learning objectives:

- Recognize emerging technologies trending in industry.
- Describe the main concepts of the emerging technologies covered in the course.
- Compare the emerging technologies discussed with their predecessors (if any)
- Understand when it is appropriate to use each of the emerging technologies learned.
- Apply the concepts learned in the design of IT solutions.

Relationship with student outcomes:

	Student Outcomes					
	1	2	3	4	5	6
Relevance	4	5			3	2

7. Topics of the course:

- The technologies/tools that will be presented in the course will vary, as trends in the industry change.