Introduction to Software Development 2023-2024 Catal

[ARCHIVED CATALOG]

SDEV 140 - Introduction to Software Development

PREREQUISITES: SDEV 120 - Computing Logic
PROGRAM: Software Development
CREDIT HOURS MIN: 3
LECTURE HOURS MIN: 2
LAB HOURS MIN: 2
DATE OF LAST REVISION: Fall, 2020

Introduces students to concepts and practices of programming languages and software development. Students are introduced to algorithms and development tools used to document/implement computer logic. Discusses the history of software development, the different types of programming such as real time processing, web/database applications, and different program development environments. Students are introduced to structured programming concepts such as basic control structures, variables, constants, arrays, procedures and functions; and advanced concepts such as lists, records, sorts and searches. Students are introduced to object-oriented software development. Concepts will be applied using different programming languages, and students will develop and test working programs in an integrated system. Students will practice skills such as team building, work ethic, communication, documentation, and adaptability.

MAJOR COURSE LEARNING OBJECTIVES: Upon successful completion of this course the student will be expected to:

1. Distinguish between systems software and application software.
2. Compare and utilize compilers, interpreters and code generators.
3. Identify variables, constants and data types used in programming.
4. Identify and use control structures.
5. Explain abstraction, modularization, functions and parameter passing in programming.
6. Explain the fundamentals of object-oriented programming concepts.
7. Write, perform use-case testing, debug and document programs in an integrated development environment.
8. Develop competence in the techniques of systematic problem analysis, algorithm development, program construction and documentation.
9. Apply the phases and design concepts of software development.
10. Gain an understanding of the basic concepts of good user-interface design.
11. Describe industry-standard software engineering tools.
12. Discuss social, legal and ethical issues in software engineering.
13. Examine basic concepts related to secure programming.
14. Examine the use of software repositories and collaboration tools in software development.

COURSE CONTENT: Topical areas of study include -

- Systems and application software
- Compilers, interpreters and code generators
- Variables, constants and data types
- Control structures
- Problem analysis and algorithm development
- Abstraction and modularization
- Functions and parameter passing
- Classes, objects, methods and inheritance

- Software development phases and models
- Software architecture and design concepts
- Project management and tracking
- Software requirements specification
- Software design interfaces
- Software testing plans and techniques
- Software engineering tools
- Input Validation
- Type Checking
- Exception Handling
- Coding Standards
- Peer Review/Collaboration

Course Addendum - Syllabus (Click to expand)
