American Public University System

The Ultimate Advantage is an Educated Mind

School:
Course Number: CSCI471
Course Name: Software Engineering
Credit Hours: 3
Length of Course: 16 Weeks
Prerequisite:

Table of Contents

Instructor Information

Instructor information available in the classroom

Table of Contents

Course Description (Catalog)

This course is an introduction to the field of modern software engineering and is geared towards solving practical problems using Python. Special attention in the course is dedicated to specifying, designing, building, testing, and delivering reliable software systems, with an additional emphasis placed on modern design techniques used for reducing complexity and increasing security of software solutions. Additionally, the content of the course examines various software development approaches by providing an overview of methods and techniques used in software development processes and contrasting agile approaches with traditional software development methods.

Table of Contents

Course Scope

Software engineering is field that aims for creating practical, cost-effective solutions to computational problems, and it does so by applying the concepts and practices of engineering and computer science to the development and maintenance of reliable, usable, and dependable software. This course covers such fundamental to software engineering topics, as understanding system requirements, modeling conceptual solutions, finding appropriate compromises, and identifying effective methods of design, coding, and testing of the code. Python is used as a primary programming language of this course. Successful practice of software engineering rests on a solid understanding of systems development life cycle- on how the phases and activities of the software development process cycle work together. Consequently, a part of the content of the course is dedicated to the presentation of traditional and agile software development methodologies.

Table of Contents

Course Objectives

Over the course of the semester students will be able to:

1. Describe, compare, and contrast various software development methods and their appropriate contexts.
2. Identify, select, and apply sound software development practices for solving real-life business problems.
3. Acquire requirements- and software-modeling skills for the purposes of traditional and object-oriented systems development.
4. Demonstrate the ability to apply sound design methods for the purposes of reducing complexity and increasing security of software systems.
5. Develop an ability to evaluate the effectiveness and efficiency of software development practices and to suggest appropriate improvements when applicable.
6. Acquire an advanced level of programming skills using Python.
7. Demonstrate an implementation of sound software development methods in a practical project.

The accomplishment of the course objectives will be assessed:

1. Critical Thinking
2. Problem Solving
3. Written Communication
4. Information Literacy (subject-matter knowledge)
5. Reading.

Table of Contents

Course Delivery Method

This course delivered via distance learning will enable students to complete academic work in a flexible manner, completely online. Course materials and access to an online learning management system will be made available to each student. Online assignments are due by Sunday evening of the week as noted and include Forum questions (accomplished in groups

through a threaded forum), examination, and individual assignments submitted for review by the Faculty Member). Assigned faculty will support the students throughout this eight-week course.

Table of Contents

Course Resources

Required Course Textbooks
Pro Python3: Features and Tools for Professional Development (3rd Edition).
J. Burton Browning and Marty Alchin
Apress Media
ISBN: 978-1-4842-4384-8

Required Reading - See Course e-reserve for required reading.

Table of Contents

Evaluation Procedures

Describe how you will evaluate your students for each graded activity.
Forum Assignments (CO1, CO2, CO3): 30%
Coding Assignments (CO4, CO5): 30%
Final Project (CO1, CO2, CO3, CO4, CO5): 40 %

Grade Instruments	Points	# in the course	% of the Grade
Assignments	**100**	**5**	**30%**
Discussion Forum Post	**100**	**16**	**30%**
6-part Course Project	**100**	**6**	**40%**
Total			**100%**

Table of Contents

16 – Week Course Outline

Please see the Student Handbook to reference the University's grading scale.

Table of Contents

Week	Topic	Learning Objectives	Assignment

1	**Principles and Philosophy**	LO-1: To demonstrate an understanding of fundamental principles and philosophies of Python (CO2, CO3, CO5) LO-2: To be able to discuss benefits and shortcomings of Python and illustrate the limits of its applicability (CO1, CO3, CO4, CO5)	Discussion Board #1
2	**Advanced Basics**	LO-1: To understand fundamental tools and techniques used in Python development (CO1, CO3) LO-2: To be able to critically evaluate some of the advanced features of Python and present types of problems that could be addressed via those features. (CO1, CO2, CO3) LO-3: To be able to implement a set of business requirements in Python (CO1, CO2, CO4)	Discussion Board #2 Coding Assignment #1
3	**Functions**	LO-1: To understand types and variety of features of functions used in Python (CO4, CO5) LO-2: To be able to communicate the benefits of some of the fundamental features of Python in regard to the implementation of the functions (XO1m CO3, CO4, CO5)	Discussion Board #3
4	**Classes**	LO-1: To understand the types and classes as the way of implementation of Object-Oriented paradigm in Python (CO4, CO5) LO-2: To be able to discuss the benefits and shortcomings of various way of implementing Object-Oriented features in Python (CO1, CO3, CO4, CO5) LO-3: To be able to implement a set of business requirements in Python (CO1, CO2, CO4)	Discussion Board #4 Coding Assignment #2
5	**Common Protocols**	LO-1: To understand the ways of implementing interfaces in Python (CO4, CO5) LO-2: To understand and be able to explain pluses and minuses of	Discussion Board #5

		operations' implementation in Python (CO1, CO3, CO4, CO5)	
6	**Object Management**	LO-1: To understand the ways of object management in Python (CO4, CO5) LO-2: To understand and be able to explain the implementation of the objects in Python (CO1, CO3, CO4, CO5) LO-3: To be able to implement a set of business requirements in Python (CO1, CO2, CO4)	Discussion Board #6 Coding Assignment #3
7	**Strings**	LO-1: To understand the implementation of built-in strings in Python (CO4, CO5) LO-2: To be able to explain the difference in the formats for implementing strings in Python (CO1, CO3, CO4, CO5)	Discussion Board #7
8	**Documentation**	LO-1: To understand and be able to utilize the tools available for describing the Python code and its features (CO4, CO5) LO-2: To be able to evaluate and communicate the applicability of documenting the code in different circumstances and for different purposes (CO1, CO3, CO4, CO5) LO-3: To be able to implement a set of business requirements in Python (CO1, CO2, CO4)	Discussion Board #8 Coding Assignment #4
9	**Testing**	LO-1: To understand the basic principles and mechanisms of unit testing in Python (CO4, CO5) LO-2: To be able to understand and communicate the benefits of modular design of programs (CO1, CO3, CO4, CO5) LO-3: To be able to understand and communicate the benefits of test-driven development (CO1, CO3, CO4, CO5)	Discussion Board #9
10	**Distribution & Sheets: a CSV framework**	LO-1: To understand the decisions and steps required for distributing the Python code (CO4, CO5) LO-2: To understand the	Discussion Board #10 Coding

		fundamentals of framework and interface design (CO4, CO5) LO-3: to be able to explain the process of creating extensible and reusable code in Python (CO1, CO3, CO4, CO5) LO-4: To be able to implement a set of business requirements in Python (CO1, CO2, CO4)	Assignment #5
11	**Software Engineering: Planning phase**	LO-1: To understand and to be able to explain activities associated with Planning phase of Systems Analysis & Design/Software Engineering (CO1, CO3, CO4, CO5) LO-2: To be able to accomplish necessary scenario-based activities of Planning phase of software engineering (CO1, CO2, CO3, CO4, CO5)	Discussion Board #11 Course Project, Part 1
12	**Software Engineering: Methods for reducing complexity in Analysis phase ("As-is" system and system requirements)**	LO-1: To be able to explain ways for reducing complexity in projects (CO1, CO3, CO4, CO5) LO-2: To understand and be able to explain the inherent limitations of analytic problem solving (CO1, CO3, CO4, CO5) LO-3: To be able to perform the necessary activities (system requirements' determination) associated with Analysis phase of software engineering (CO1, CO2, CO3, CO4, CO5)	Discussion Board #12 Course Project, Part 2
13	**Software Engineering: Methods for reducing complexity in Analysis phase ("To-be" system' modeling)**	LO-1: To be able to discuss fundamental purposes of the analysis of the context (CO1, CO3, CO4, CO5) LO-2: To be able to understand and communicate the fundamentals of phased approach to development (CO1, CO3, CO4, CO5) LO-3: To be able to perform the necessary activities associated with conceptual system modeling required for Analysis phase of software engineering project (CO1, CO2, CO3, CO4, CO5)	Discussion Board #13 Course Project, Part 3

14	**Software Engineering: Design phase and Security Design Principles**	LO-1: To be able to understand and communicate fundamental criteria for assessing quality of design (CO1, CO3, CO4, CO5) LO-2: To be able to explain the reasons for having tradeoffs in design (CO1, CO3, CO4, CO5) LO-3: To be able to create a modular design of a software system and to be able to design a secure module of a system based on the best practices of secure software engineering (CO1, CO2, CO3, CO4, CO5)	Discussion Board #14 Course Project, Part 4
15	**Software Engineering: Implementation phase**	LO-1: To describe the activities involved in Implementation phase of software engineering (CO3, CO4, CO5) LO-2: To be able to write a robust code to implement a scenario-based functionality (CO1, CO2, CO3) LO-3: To be able to write tests to assess the robustness and security of the program (CO1, CO2, CO3)	Discussion Board #15 Course Project, Part 5
16	**Software Engineering: Closing the project**	LO-1: To be able to prepare and present the content of a large project in a professional manner (CO1, CO2, CO3) LO-2: To be able to design and prepare system documentation (CO1, CO2, CO3) LO-3: To be able to able to design and prepare system user training manual (CO1, CO2, CO3)	Discussion Board #16 Course Project, Part 6

Table of Contents

Policies

Communications

Student Communication

To reach the instructor, please communicate through the MyClassroom email function accessible from the Classlist of the Course Tools menu, where the instructor and students email addresses are listed, or via the Office 365 tool on the Course homepage.

- In emails to instructors, it's important to note the specific course in which you are enrolled. The name of the course is at the top center of all pages.
- Students and instructors communicate in Discussion posts and other learning activities.
- All interactions should follow APUS guidelines, as noted in the Student Handbook, and maintain a professional, courteous tone.
- Students should review writing for spelling and grammar.
- Tips on Using the Office 365 Email Tool

Instructor Communication

The instructor will post announcements on communications preferences involving email and Instant Messaging and any changes in the class schedule or activities.

- Instructors will periodically post information on the expectations of students and will provide feedback on assignments, Discussion posts, quizzes, and exams.
- Instructors will generally acknowledge student communications within 24 hours and respond within 48 hours, except in unusual circumstances (e.g., illness).
- The APUS standard for grading of all assessments (assignments, Discussions, quizzes, exams) is seven days or fewer from the due date.
- Final course grades are submitted by faculty no later than seven days after the end date of the course or the end of the extension period.

University Policies

Consult the Student Handbook for processes and policies at APUS. Notable policies:

- Drop/Withdrawal Policy
- Extension Requests
- Academic Probation
- Appeals
- Academic Dishonesty / Plagiarism
- Disability Accommodations
- Student Deadlines
- Video Conference Policy

Mission

The mission of American Public University System is to provide high quality higher education with emphasis on educating the nation's military and public service communities by offering respected, relevant, accessible, affordable, and student-focused online programs that prepare students for service and leadership in a diverse, global society

Minimum Technology Requirements

- Please consult the catalog for the minimum hardware and software required for undergraduate and graduate courses.
- Although students are encouraged to use the Pulse mobile app with any course, please note that not all course work can be completed via a mobile device.

Disclaimers

- Please note that course content – and, thus, the syllabus – may change between when a student registers for a course and when the course starts.
- Course content may vary from the syllabus' schedule to meet the needs of a particular group.

Turnitin.com

Faculty may require assignments be submitted to Turnitin.com. Turnitin.com will analyze a paper and report instances of potential plagiarism for the student to edit before submitting it for a grade. In some cases professors may require students to use Turnitin.com.

Table of Contents