

NCBS 443A: JUST-IN-TIME SUPPORT FOR C S 3A

Foothill College Course Outline of Record

Heading	Value
Effective Term:	Summer 2022
Units:	0
Hours:	2.5 lecture per week (30 total per quarter)
Corequisite:	C S 3A.
Degree & Credit Status:	Non-Degree-Applicable Non-Credit Course
Foothill GE:	Non-GE
Transferable:	None
Grade Type:	Non-Credit Course (Receives no Grade)
Repeatability:	Unlimited Repeatability

Student Learning Outcomes

- Students are able to discuss and apply debugging techniques, and use the debugger included in their integrated development environment
- Students are able to discuss and apply PEP-8 and other python style guidelines
- Students will demonstrate an ability to read project specifications and write functioning programs that meet those specifications
- Students will demonstrate the ability to install an Integrated Development Environment on a computer

Description

A just-in-time approach to the core prerequisite skills, competencies, and concepts needed in C S 3A. Intended for students who are concurrently enrolled in C S 3A at Foothill College. Topics include: installation of an integrated development environment and other software, navigating a file system hierarchy, developing a logic-based approach to programming, identifying errors in a program using a debugger and other means.

Course Objectives

The student will be able to:

1. Explore topics related to developing effective learning skills
2. Install integrated development environment software
3. Manipulate a hierarchical file system
4. Write code that follows a software specification/requirements document
5. Demonstrate an understanding of flow control using flowcharts and other means
6. Identify and fix program errors using a debugger and other means
7. Write pseudocode and turn pseudocode into programming code
8. Follow style conventions in a particular programming language

Course Content

1. Explore topics related to developing effective learning skills
 - a. Learn study skills
 - b. Organizational skills
 - c. Time management
 - d. Test preparation
 - e. Research
2. Install integrated development environment software
 - a. Navigate to a vendor site and choose an appropriate operating system and software version
 - b. Unpack software as needed
 - c. Choose appropriate installation options
 - d. Solve installation issues
3. Manipulate a hierarchical file system
 - a. Navigate to a target folder/directory
 - b. Move, copy, delete and rename files
4. Write code that follows a software specification/requirements document
 - a. Parse the spec into required program elements, such as classes, functions, and variables
 - b. Run provided testing code to verify that a program behaves as expected
 - c. Develop testing code to verify that a program meets spec
 - d. Prepare a sample run to document successful testing
5. Demonstrate an understanding of flow control using flowcharts and other means
 - a. While loops
 - b. For loops
 - c. If statements
 - d. Exit conditions
6. Identify and fix program errors using a debugger and other means
 - a. Unconditional and conditional breakpoints
 - b. Watch lists
 - c. Stack trace
7. Write pseudocode and turn pseudocode into programming code
8. Follow style conventions in a particular programming language

Lab Content

Not applicable.

Special Facilities and/or Equipment

Access to a computer laboratory with the appropriate software.

Method(s) of Evaluation

Methods of Evaluation may include but are not limited to the following:

Group and independent exploratory activities
Homework
Performance in C S 3A

Method(s) of Instruction

Methods of Instruction may include but are not limited to the following:

Group work
Discussion
Mini-lectures
Instructor-guided discovery
Formative assessment

Representative Text(s) and Other Materials

Horstmann, Cay S., and Rance D. Nicaise. [Python for Everyone, 3rd ed.](#). 2019.

Types and/or Examples of Required Reading, Writing, and Outside of Class Assignments

1. [Assigned reading from the parent course, and supplemental reading as assigned to reinforce course concepts](#)
2. Written documentation of code
3. Written reflection after completing an assignment, and after receiving feedback
4. Supplemental coding assignments to reinforce concepts from the parent course

Discipline(s)

Computer Science