ITP 100 (D01A)

Software Design

Course Syllabus

Instructor	Jeffrey Elkner
Session	Fall 2019
Meeting Days	B Day
Time	11:05 am - 12:30 pm or 1:40 - 3:10 pm
Location	Arlington Career Center Room
Contact	jeffrey.elkner@apsva.us / jde232@email.vccs.edu

Course Description:

Introduces principles and practices of software development. Includes instruction in critical thinking, problem solving skills, and essential programming logic in structured and object-oriented design using contemporary tools.

General Course Purpose:

In this course students will learn to design software using both structured programming and object-oriented programming concepts. Students will be introduced to pseudo-code, flow charts, and UML, and will learn to program in Python.

Course Prerequisites/Co-Requisites:

Prerequisite: Placement in MTH 154 or higher.

Course Objectives:

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

- Identify programming terminology and basic mechanics of programming necessary for success in programming courses.
- Describe the structured design building blocks applied within programming solutions.
- Illustrate structured design using an appropriation notational language (such as pseudo code and/or flow charting)
- Design the core concepts of object-oriented design
- Illustrate O-O designs using the standard Unified Modeling Language using an appropriate design tool

ITP 100 (D01A)
Software Design
Course Syllabus

Major Topics to be Included

Program Design

- Program analysis and design within the system development life cycle.
- Evolution and development of both programming languages and program design.
- Difference between a design notational language and a design tool.
- Development of program algorithms using sequence, selection, and looping tools

Structured Design

- Design process for structured program design.
- Notational languages (pseudo code and/or flow charting) and design tools for structured design.
- Sequence, selection, and loop structures within a structured design solution for an operation.
- Decisions using null ELSE selections, nested selections, and CASE structures within a structured design solution.
- WHILE, UNTIL, FOR loops, and nested loops within a structured design
- Single dimensional and parallel arrays
- Modular code using predefined programming functions.
- Modular code with user written programming functions.
- Structured design solutions that involve one operation calling other operations with received and returned arguments.

Object-Oriented Design

- Design process for object-oriented program design.
- Unified Modeling Language (UML) within object oriented design.
- Tools for standard O-O notational language, the Unified Modeling Language.
- Candidate classes given a problem description.
- Attributes and methods for candidate classes given a problem description.
- Operations with a full UML signature including received and returned arguments.
- Object, class, message, data-hiding, information-hiding, encapsulation concepts.
- UML class diagrams.

Cooperative Teamwork

- Design team approach to software design

ITP 100 (D01A)
Software Design
Course Syllabus

Student Learning Outcomes

Program Design

- Relate the place of program analysis and design within the system development life cycle.
- Describe the evolution and development of both programming languages and program design.
- Describe the difference between a design notational language and a design tool.
- Develop program algorithms.

Structured Design

- Describe a reasonable design process for structured program design.
- Identify one or more appropriate notational languages (such as pseudo code and/or flow-charting) and tools for structured design.
- Illustrate sequence, selection, and loop structures within a structured design solution.
- Illustrate decisions using null ELSE selections, nested selections, and CASE structures within a structured design solution for an operation.
- Demonstrate a working knowledge of when to apply the appropriate loop structure within a structured design.
- Illustrate WHILE, UNTIL, FOR loops, and nested loops within a structured design solution for an operation.
- Illustrate single dimensional and parallel arrays within a structured design solution.
- Illustrate modular code using predefined programming functions.
- Illustrate modular code with user written programming functions.
- Create structured design solutions that involve one operation calling other with received and returned arguments.

Object-Oriented Design

- Describe a reasonable design process for object-oriented program design.
- Explain the use and importance of the Unified Modeling Language within object oriented design.
- Identify appropriate tools to use with the standard O-O notational language, the Unified Modeling Language.
- Identify appropriate candidate classes given a problem description.
- Identify appropriate attributes and operations for candidate classes given a problem description.
- Show operations with a full UML signature including received and returned arguments.
- Define the terms object, class, message, data-hiding, information-hiding, encapsulation.
- Create a UML class diagram.

Cooperative Teamwork

- Operate as a member of a design team on a given design task.
- Design a single task of a project while other teams or individuals are working on separate tasks of the same project.

ITP 100 (D01A)
Software Design
Course Syllabus

Time Allocation per Topic

The following table has the breakdown of the time in this course that will be spent on each course topic:

Topic	Hours	Percentage
Introductory Programming Concepts	3	6.5%
Structured Program Design	3	6.5%
Decisions	6	13%
Loops	7.5	17%
Arrays	7	16%
Functions and Modular Programming	7	16%
Object Oriented Programming	7	16%
Other Optional Content	2.5	5%
Exams and Quizzes (NOT including final exam)	2	4%
Total:	**45**	**100%**

Required Instructional Materials:

- Getting Down with the Unix CLI, *by Jeffrey Elkner*
- CS Principles: Big Ideas in Programming *by Mark Guzdial and Barbara Ericson*
- Algorithms to Live By: The Computer Science of Human Decisions *by Brian Christian and Tom Griffiths*
- GASP Python Course

Course Credit: 3 Credits

ITP 100 (D01A)

Software Design

Course Syllabus

Policies:

I. Expectations

a. Software Design is a rigorous, college level course that will require sustained and consistent engagement from students.

b. An average of 90 minutes of homework will be assigned for each 90 minutes in class. We will be utilizing a flipped classroom learning environment, where the lecture portion of the course material will be viewed individually at home *before* class meets, and class time will be used for collaborative engagement and discussion.

c. Daily "mini quizzes" at the beginning of class will be used to be sure homework readings and practice have been completed. To be successful in this class, students will be expected to be prepared for these quizzes when they arrive in class.

II. Grading Policies

a. Grading Scale
A= 100 - 90 B= 89 - 80 C= 79 - 70 D=69 - 60 F= 59 and below

b. Students will receive a weekly cumulative letter grade that will incorporate daily quizzes, tests, projects, and presentations. These weekly evaluations can be challenged by the student, *but only during the week immediately following when the evaluation is given*.

c. The average of the weekly evaluations will make up 70% of the final grade, with the course final exam making up 30%.

d. In cases where district grading policies conflict with college grading policies, the high school and college grades may differ; this may include assignment/test retakes, extended assignment due dates, capped minimum grade allowed, among other such district policies.

e. It is important that students check their final NOVA grades in Blackboard as soon as the course(s) completed.

III. Course Policies

a. Academic Integrity

i. The College does not tolerate academic dishonesty. Students who are not honest in their academic work will face disciplinary action along with any grade penalty the instructor imposes. Procedures for disciplinary measures and appeals are outlined in the Student Handbook (http://www.nvcc.edu/students/handbook/). In extreme cases, academic dishonesty may result in dismissal from the College.

ii. Plagiarism: is the act of appropriating passages from the work of another individual, either word for word or in substance, and representing them as one's own work. This includes any submission of written work other than one's own. In short, plagiarism means using the exact words, opinions, or factual information from another person without giving that person credit. Students who are not honest in their academic work will face disciplinary action along with any grade penalty the instructor imposes. For more information about student academic integrity: https://www.nvcc.edu/curcatalog/policies/integrity.html

b. Disabilities

i. Students with disabilities are required to contact NOVA's Office of Disability Support Services (DSS) to discuss possible accommodations. All information is kept confidential and may increase your chances of success in the academic setting. If accommodations are agreed upon, student will receive a Memorandum of Accommodation (MOA) by DSS. For more information about NOVA's DSS office: https://www.nvcc.edu/disability-services.

ITP 100 (D01A)
Software Design
Course Syllabus

c. Self-Advocacy

i. Students are expected to reach out to their instructor if they do not understand content or expectations.

ii. College instructors and other college personnel will not talk with a parent without the permission of and presence of the student. The conversation is between the administrator / faculty member and the student. The parent's role is to listen, give moral support, and summarize information and agreements if needed.

iii. Dual enrolled students have access to full NOVA campus services including tutoring, library, and counseling services; student resources are found here: http://www.nvcc.edu/students/index.html

IV. Course Schedule

a. Critical Course Dates

Course Start Date	Monday, September 16, 2019
Course Drop Date	Friday, October 4, 2019
Course Withdrawal Date	Friday, December 6, 2019
Final Exam Date	Week of January 27 to 30, 2020
Course End Date	Thursday, January 30, 2020

b. Final Exam Date: *The final exam will be given during the last week of class, between Monday, January 27 and Friday, January 30.*