Revised 8/2023

NOVA COLLEGE-WIDE COURSE CONTENT SUMMARY
ITP 250 – ADVANCED PHYTON PROGRAMMING (4 CR.)

Course Description

Object-oriented design and advanced programming concepts using Python through instruction and hands-on programming. Emphasizes Object-oriented design (OOD) Object Orient Programming (OOP) concepts, such as classes, inheritance, polymorphism, Object-oriented design patterns, and Unified Modeling Language (UML). Examines best practices, code reusability, and exploration of Python modules and advanced topics.
Lecture 4 hours. Total 4 hours per week.

General Course Purpose

This course provides instruction and hands-on programming requirements for a student to create, modify, test, and debug Python programs that contain object-oriented constructs and advanced Python components.

Course Prerequisites/Corequisites

Prerequisite: ITP 150.

Course Objectives

Upon completing the course, the student will be able to:

- Design, code, test, and debug Python programs in a hands-on approach using an Integrated Development Environment (IDE) to implement concepts covered in this course
- Apply foundational knowledge of Python by writing code which illustrates a full range of object-oriented constructs (OOP) and illustrates more advanced features of the Python language
  - Write code using the Python Standard Library
  - Write Python code that calls built-in functions, user-defined functions, and methods.
  - Write Python code that has/uses each of the foundation constructs:
  - Write Python code that uses a main function
- Import and write code that uses various Python modules and/or third-party modules
- Perform OOD tasks using UML. Implement the design using OOP constructs and OO best practices
- Explain exceptions their uses, and their benefits.
- Explain various OO concepts, including classes, objects, encapsulation, and data hiding
- Explain the relationships involved with inheritance and its benefits. Explain abstract methods, abstract classes, and polymorphism.
- Create one or more UML OOD class diagrams that illustrate the following:
- Write code that illustrates code reusability. Explain the benefits of code reuse and benefits of code maintenance considerations.
- Use an Integrated Development Environment (IDE) to write, update, and test Python code
- Explain the difference between immutable and mutable data types.
- Identify Python built-in data types.
- Work with text files

Major Topics to be Included

- Hands-on programming using a Python Integrated Development Environment (IDE)
- Continued programming and advanced use of various foundational constructs, including:
- Object-oriented design and object-oriented programming, including:
- Recursion and advanced function / method code
- Best practices regarding code reusability, reduction of code maintenance, and testing
- Third-party modules
- Exploration of additional advanced topics using Python. May include additional Python modules, common frameworks, and/or advanced programming concepts.
- Functions
- Other Python language features