

ITM 313 SYLLABUS

ITM 313 Introduction to Open Source Application Development

Hours: 3 credit hours / 60 contact hours; 30 hours lecture, 30 hours lab

Instructor: Sheikh “Sam” Shamsuddin

Textbook, title, author, and year: *Introduction to Programming Using Python*, Daniel Y., 2013.

Specific course information

- a. **Catalog description:** Introduces basic concepts of systems programming using a modern open source language. Students learn to apply basic programming concepts toward solving problems, writing pseudocode, working with and effectively using basic data types, abstract data types, control structures, code modularization and arrays. They will learn to detect errors, work with variables and loops, and discover how functions, methods, and operators work with different data types. Students will be introduced to the object paradigm including classes, inheritance, and polymorphism.
- b. **Prerequisites:** None
- c. **Required**

Specific goals for the course

- a. **Program Educational Objectives:**
 2. Perform requirements analysis, design and administration of computer and network-based systems conforming to policy and best practices, and monitor and support continuing development of relevant policy and best practices as appropriate.
- b. **Course Outcomes:**
 - Gain experience skills to write, compile, execute, troubleshoot, analyze, evaluate, and resolve simple problems through program coding using Python computer language.
 - Acquire the ability to develop, synthesize, and identify important language standard libraries and utilities.
 - Build experience in software application and development theory and concepts
- c. **Course Student Outcomes:**

Upon successful completion of the course the student should be able to:

 - Recall and describe software application and development theory and concepts
 - Write, compile, execute, troubleshoot, analyze, evaluate, and resolve simple problems through program coding using Python computer language.
 - Develop, synthesize, and identify important language standard libraries and utilities.
 - Apply data transfer techniques between modules using parameters and return values.
 - Construct applications to use simple files for input and output.
 -

- Implement arrays as structures to contain data.
- Use a higher-level programming language to code, test, and debug software designs.
- Implement concepts of Object Oriented Programming (OOP), inheritance and polymorphism.
- Describe integration of Graphical User Interfaces (GUIs) and event driven programming.
- Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions **(ABET Computing Criterion 3.1)**
- Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline **(ABET Computing Criterion 3.2)**
- Identify and analyze user needs and take them into account in the selection, creation, evaluation, and administration of computer-based systems **(ABET IT Criterion 3.6)**

Topics to be covered

- a. Data Types. Arithmetic operation. Variable declarations. Flowcharts, Pseudo Codes, Simple programming. Software Development.
- b. Learning a programming language. Input/Output. Decision Structure.
- c. Continue on Decision Structure and Flowcharting.
- d. Repetition Structure (looping).
- e. Continue on Repetition Structure. Pseudo codes, Flowcharts.
- f. Debugging programs
- g. Functions and flowchart Array/List File I/O
- h. Objects and classes Object and classes – private data fields
- i. Inheritance
- j. Graphical User Interface (GUI).