**Based on ABET ETAC Student Learning Outcomes**

1. **Course Number and Name:**
   CSET 1100 Introduction to Computer Science & Engineering Technology

2. **Credits and Contact hours:**
   Credits: 4 hours, Contact: 3 lecture hours, 1 lab hour

3. **Instructor's or course coordinator's name:**
   Jared Oluoch

4. **Text book, title, author, and year:**
   Introduction to Programming Using Python, Y. Daniel Liang, 2012
   a. **Other supplemental materials:**
      None

5. **Specific Course Information:**
   a. **Brief description of the content of the course (catalog description):**
      This three semester hour course is the first course in computer hardware and software for CSET majors. Single and multi-user operating systems, command-line processing, program planning and creation and simple Internet tools are covered.
   b. **Pre-requisites, or co-requisites:**
      MATH 1330

6. **Specific goals for the course:**
   a. **Specific outcomes of instruction:**
      1. Describe the fundamentals of computer history, hardware, software and OS.
      2. Connect from a remote location to a UNIX server on which they have an established shell account.
      3. Transfer files to and from a PC or Mac to their account on a UNIX server.
      4. Use the UNIX "man" pages to determine the proper syntax for user-level UNIX commands.
      5. Manage their file system on a UNIX server.
      6. Create technical and simple game programs using the Python programming language.
      7. Debug and test programs.
      8. Describe issues related to user interfaces and user friendliness.
      9. Describe and implement the data structures available in the Python programming language.
      10. Design simple algorithm using pseudocode and/or a flowchart.
      11. Create programs that implement a variety of common algorithms using the Python programming language.

b. **Explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course: 1, 2**
   1. An ability to apply knowledge, techniques, skills and modern tools of mathematics, science, engineering, and technology to solve broadly defined engineering problems appropriate to the discipline;
   2. An ability to design systems, components, or processes meeting specified needs for broadly defined engineering problems appropriate to the discipline.

7. **Brief list of topics to be covered:**
   1. History of Computing
   2. Overview of OS (Unix as example)
   3. Introduction to net centric computing (FTP,SSH etc)
   4. Machine level representation of data
   5. Assembly level machine organization
   6. Overview of Programming Languages
   7. Procedure Development: Pseudocode, Flowcharting & Control Structures
   8. The process and mechanism of creating a program
   9. Program environment, tools
   10. Intro to Python: Basic Programming structure
   11. Basic Syntax, Variable declaration, I/O
   12. Basic Operators, Expression evaluation, Priority, casting
   13. Introduction to the gcc for compiling C program code
   14. Block statement, Python Flow Control: Branching
   15. Looping, break, continue, sentinel
   16. Abbreviations, Characters conversion
   17. Debugging and testing, error and handling
   18. Library, Function
   19. Scope of variables, parameter passing
   20. One-Dimensional Arrays, searching, summing, etc.
   21. String and operations
   22. Properties of good software design, user friendliness
   23. Higher-Dimensional Arrays
   24. Typedef, structures
   25. Text File Processing
   26. Software requirements, validation and verification
   27. Problem solving using algorithm, simple algorithm
   28. Sequential and binary search
   29. Quadratic Sorting algorithms
   30. O(N log N) sorting