CS100
Introduction to Programming
[Onsite]

Course Description:

This course serves as a foundation for understanding the logical function and process of computer programming in a given language environment. Basic computer programming knowledge and skills in logic and syntax will be covered. Coding convention and procedures will be discussed relevant to the given programming language environment.

Prerequisite(s) and/or Corequisite(s):

Prerequisites: TB143 Introduction to Personal Computers or equivalent

Credit hours: 4

Contact hours: 50 (30 Theory Hours, 20 Lab Hours)

Syllabus: Introduction to Programming

Instructor:	_____
Office hours:	_____
Class hours:	_____

Major Instructional Areas

1. Fundamental concepts of computer programming
2. Memory allocation and variables
3. Problem specification and analysis using flowcharting and pseudocode
4. Conditional statements and repetition structures
5. Coding conventions and procedures

Course Objectives

1. Describe the fundamental concepts in computer programming.
2. Create basic software program designs.
3. Create various modules in computer programs.
4. Create various functions in computer programs.
5. Create computer programs that can make decisions.
6. Create computer programs that can do repetitive processing.
7. Create computer programs that can do input validation.
8. Create program software that incorporates modules (and/or functions), conditional logic, looping, and input validation.
9. Test program software.

SCANS Objectives

SCANS is an acronym for Secretary's Commission on Achieving Necessary Skills. The committee, created by the National Secretary of Labor in the early 1990s, created a list of skills and competencies that the committee feels are necessary for employees to function in a high-tech job market.

1. Identify relevant facts and analyze information in a logical manner after locating and verifying information using resources and computers.
2. Identify common goals and examine all possible options for problem solving.
3. Identify problems, create and implement solutions, and revise solutions, as required.
4. Allocate time and energy for completing projects in a timely manner.
5. Exert a high level of effort and perseverance toward attaining goals.
6. Recognize problems and devise and implement a plan of action.
7. Demonstrate the ability to utilize authentic resources available, including the Internet, knowledge libraries, or other sources.
8. Locate, understand, and interpret information obtained from a variety of sources.
9. Identify the need for data; select, retrieve, and analyze information; and communicate the results of information analysis in written, graphical, and pictorial formats.
10. Compare and contrast two theories or alternatives to arrive at the best solution.
11. Apply procedures, tools, and equipment—including computers and related technologies—whenever required.
12. Evaluate alternatives and choose the best for a situation.

Course Outline

Note: All graded activities, except the Lab Practicum and Unit Exams, are listed below in the pattern of <Unit Number>.<Assignment Number>. For example, Lab 1.1 refers to the 1st lab activity in Unit 1.

Unit	Activities
1– Fundamental Concepts	• Content Covered: *Starting Out with Programming Logic & Design* o Chapter 1, "Introduction to Computers and Programming" • Labs: 1.1 • Assignments: 1.1
2– Software Program Design	Read from *Starting Out with Programming Logic & Design:* o Chapter 2, "Input, Processing, and Output," pp. 29-55 • Labs: 2.1 • Assignments: 2.1
3– Software Program Design (con't.)	Read from *Starting Out with Programming Logic & Design:* o Chapter 2, "Input, Processing, and Output," pp. 56-68 • Labs: 3.1 • Assignments: 3.1
4– Program Modules	Read from *Starting Out with Programming Logic & Design:* o Chapter 3, "Modules," pp. 75-90 • Unit Exams: 1 • Labs: 4.1 • Assignments: 4.1
5– Program Modules and Functions	Read from *Starting Out with Programming Logic & Design:* o Chapter 3, "Modules," pp. 90-108 o Chapter 6, "Functions," pp. 217-218, pp. 225-231 • Labs: 5.1

Unit	Activities
	• Assignments: 5.1
6– Decisions I	Read from *Starting Out with Programming Logic & Design:* o Chapter 4, "Decision Structures and Boolean Logic," pp. 115-142 • Unit Exams: 2 • Labs: 6.1 • Assignments: 6.1
7– Decisions II	Read from *Starting Out with Programming Logic & Design:* o Chapter 4, "Decision Structures and Boolean Logic," pp. 142-155 • Labs: 7.1 • Assignments: 7.1
8– Repetitive Processing I	Read from *Starting Out with Programming Logic & Design:* o Chapter 5, "Repetition Structures," pp. 163-183 and pp. 196-201 • Labs: 8.1 • Assignments: 8.1
9– Repetitive Processing II	Read from *Starting Out with Programming Logic & Design:* o Chapter 5, "Repetition Structures," pp. 183-211 • Labs: 9.1 • Assignments: 9.1
10– Repetitive Processing III and Comprehensive Lab	Read from *Starting Out with Programming Logic & Design:* o Chapter 7, "Input Validation" • Lab Practicum (Part I)
11– Course Review and	• Course Review

Unit	Activities
Comprehensive Lab	• Lab Practicum (Part II) • Unit Exams: 3

Instructional Methods

The curriculum is designed to promote a variety of teaching strategies that support the outcomes described in the course objectives and that foster higher cognitive skills. Delivery makes use of various media and delivery tools in the classroom.

Multiple styles, such as lectures, collaborative learning options, and hands-on laboratory activities, will be used to deliver content and inspire and engage students. Your progress will be regularly assessed using various accessible methods and tools. *This course uses various learning strategies such as exams, assignments, lab exercises, and a lab practicum to help you understand the concepts. Assignments are based on the concepts covered in different units. Units 4, 6, and 11 each have an exam. These exams will analyze your learning and help you recall the concepts already taught.*

Classroom practices will create a climate of high values with respect to both diversity and inclusiveness. An open communication environment will help to ensure useful interactions between students and the instructor and among students themselves. Lesson plans, course materials, notes, or other information resources will be made available and be made flexible to all students as needed.

Instructional Materials and References

Student Textbook Package

- Gaddis, Tony. *Starting Out with Programming Logic & Design. Custom Edition. Indianapolis:* Pearson Custom Publishing, 2008.
- Gaddis, Tony. *Lab Manual to Accompany Starting Out with Programming Logic & Design.* Indianapolis: Pearson Custom Publishing, 2008.
- Python/Raptor software CD (included with textbook)

References

ITT Tech Virtual Library

Log on to the ITT Tech Virtual Library at http://www.library.itt-tech.edu/ to access online books, journals, and other reference resources selected to support ITT Tech curricula.

Books

You may click “Books” or use the “Search” function on the home page to find the following books.

ITT Tech Virtual Library> Main Menu> Books> Books24x7>

- Dawson, Michael. *Python Programming for the Absolute Beginner. Boston:* Premier Press, 2003.
- Goodliffe, Peter. *Code Craft: The Practice of Writing Excellent Code. San* Francisco: No Starch Press, 2007.
- Hetland, Magnus Lie. *Practical Python. Berkeley, CA: Apress, 2002.*
- Shasha, Dennis E. *Puzzles for Programmers and Pros. Indianapolis: Wiley* Publishing, Inc., 2007.

Other References

The following resources may be found **outside** of the ITT Tech Virtual Library, whether online or in hard copy.

Web sites

- Algorithms

http://courses.cs.vt.edu/~csonline/Algorithms/Lessons/index.html

Self-paced lessons on understanding and using algorithms

- Number Systems

http://courses.cs.vt.edu/~csonline/NumberSystems/Lessons/index.html

Self-paced lessons on understanding and using number systems, including binary, hexadecimal, and octal

- Programming Languages

http://courses.cs.vt.edu/~csonline/ProgrammingLanguages/Lessons/index.html

Self-paced lessons on understanding and using various programming languages

- Programming Tutorials

http://www.programmingtutorials.com/python.aspx

A collection of links to online tutorials available on Python software

- Python

http://www.python.org

The official Web site of the Python Software Foundation, with news, documentation, and downloads

Books

- Downey, Allen B., Jeffrey Elkner, and Chris Meyers. *How to Think Like a Computer Scientist: Learning with Python. Needham, MA: Green Tea Press, 2002.*

Available on FreeTechBooks.com:
http://freetechbooks.com/about134.html&highlight=allen+downey

This downloadable book teaches readers to think like a computer scientist regarding the use of Python.

All links to Web references outside of the ITT Tech Virtual Library are always subject to change without prior notice.

Course Evaluation and Grading

Evaluation Criteria Table

The final grades will be based on the following categories:

CATEGORY	WEIGHT
Labs	30%
Assignments	15%
Unit Exam 1	15%
Unit Exam 2	10%
Unit Exam 3	15%
Lab Practicum	15%
Total	**100%**

Note: Students are responsible for abiding by the Plagiarism Policy.

Grade Conversion Table

The final grades will be calculated from the percentages earned in the course, as follows:

A	90–100%	4.0
B+	85–89%	3.5
B	80–84%	3.0
C+	75–79%	2.5
C	70–74%	2.0
D+	65–69%	1.5
D	60–64%	1.0
F	<60%	0.0

