ITT Technical Institute

PT1420T

Introduction to Programming

Onsite Course

SYLLABUS

Credit hours: 4.5

Contact/Instructional hours: 67 (41 Theory Hours, 26 Lab Hours)

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

Prerequisites: NT1110T Computer Structure and Logic or equivalent

Course Description:

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

Where Does This Course Belong?

This course is required for the Network Systems Administration and Mobile Communications Technology programs.

The Network Systems Administration program covers the following core areas:

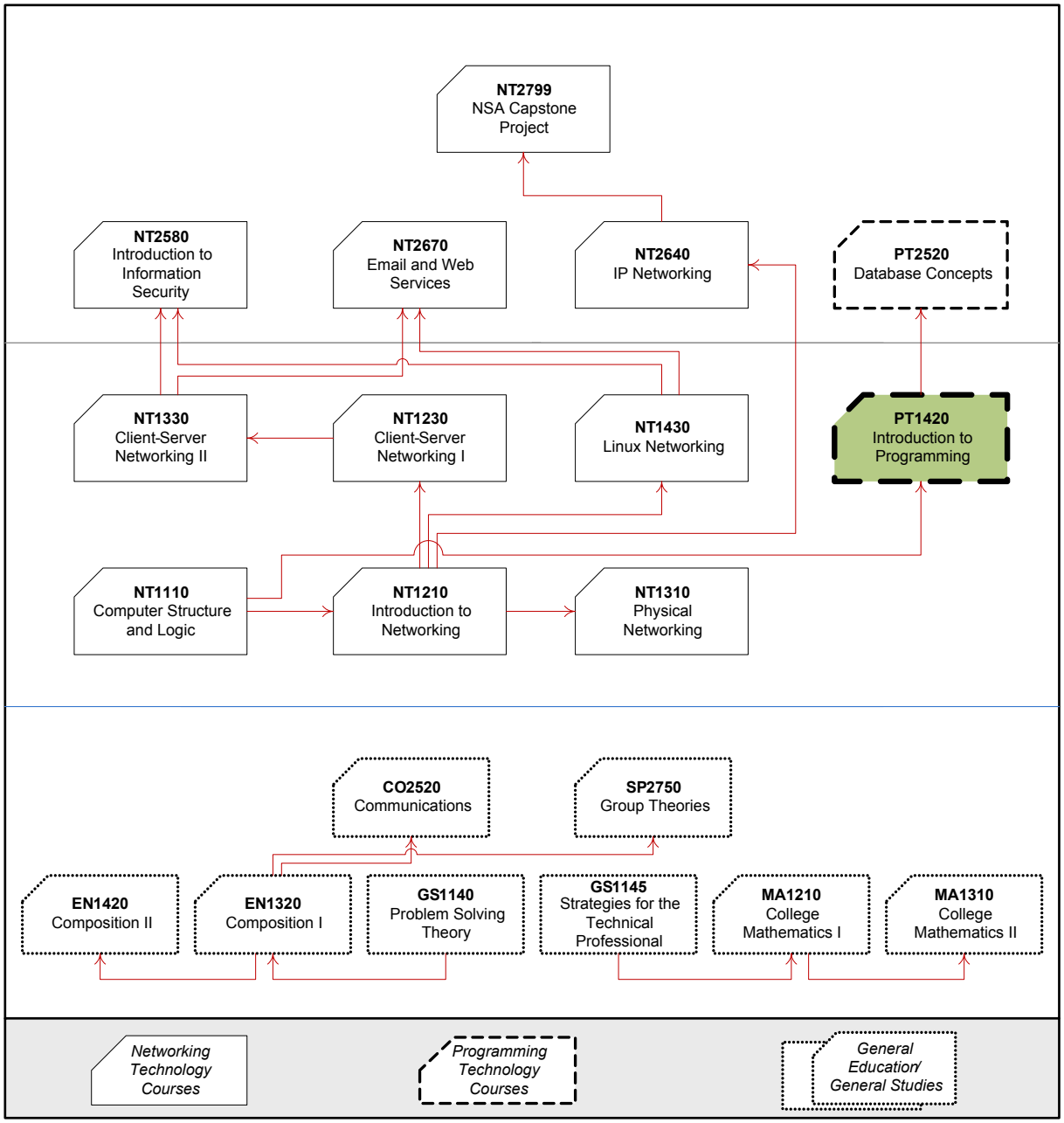
- Computer and Networking Foundation
- Networking Standards and Protocols
- Server -Client Networking
- LAN/WAN Internetworking
- Network Services and Management
- Network Security and Management
- Programming Technology
- General Education

The Mobile Communications Technology program covers the following core areas:

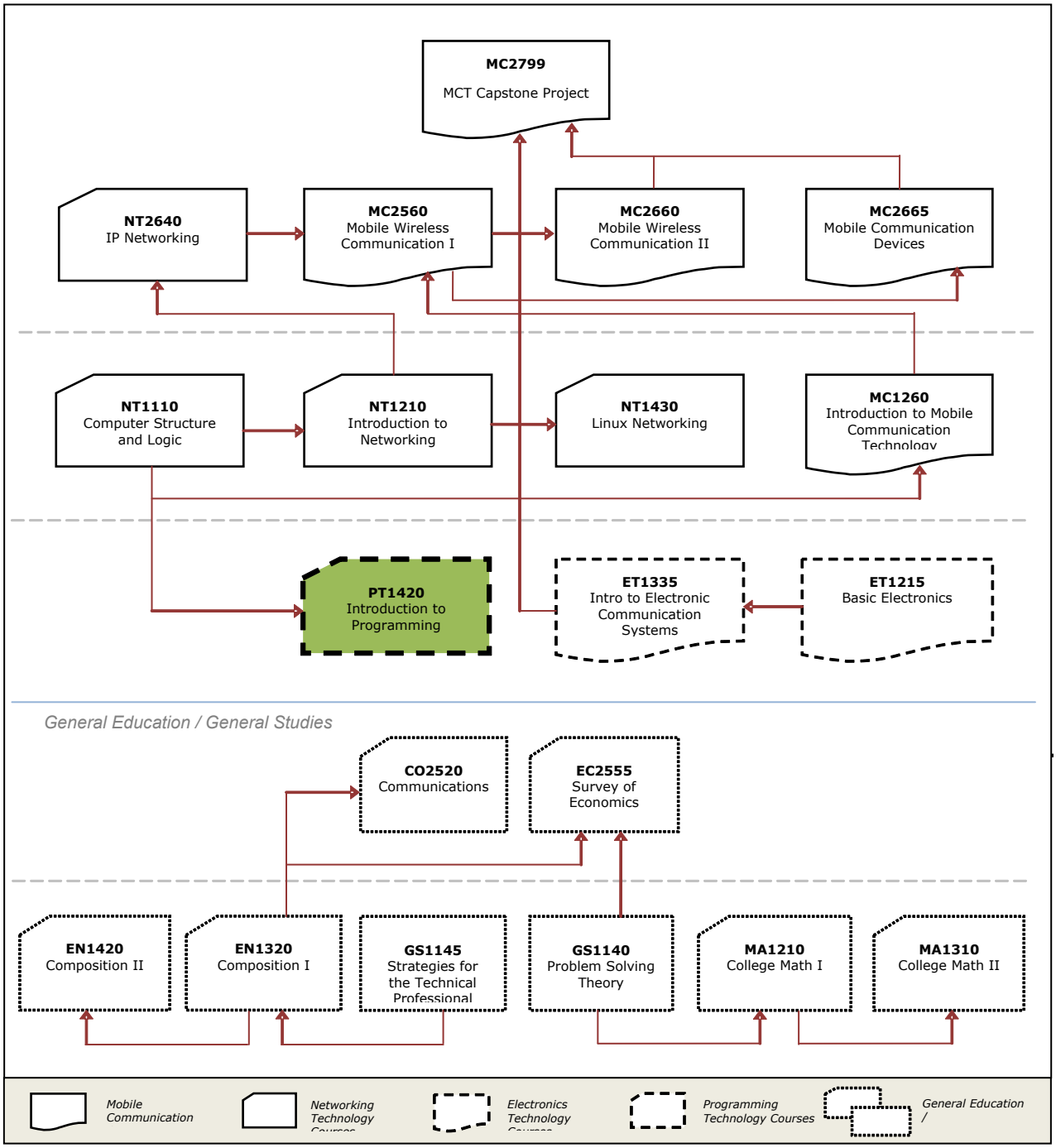
- Basic Electronics and Electronics Communication
- Networking
- Programming
- Mobile Communications Technology
- General Education

The following diagrams demonstrate how this course fits in the programs:

Associate Program in Network Systems Administration



Associate Program in Mobile Communications Technology



NOTE: *Refer to the catalog for the state-specific course information, if applicable.*

Course Summary

Major Instructional Areas

1. Program structure and reserved keywords
2. Variables and data types
3. Error handling and debugging
4. Modules
5. Conditional structures
6. Loops
7. Functions
8. Files

Course Objectives

1. Describe the fundamental concepts in computer programming.
2. Design programs by using flowcharts and pseudo code.
3. Write programs that perform input, processing, and output.
4. Write programs that use variables and constants.
5. Apply modular programming techniques to programming.
6. Write programs that use conditional statements to solve problems.
7. Write programs that use loops to solve problems.
8. Apply techniques to write functions in programs.
9. Write programs that can read data from and write data to files.

Learning Materials and References

Required Resources

Textbook Package	New to this Course	Carried over from Previous Course(s)	Required for Subsequent Course(s)
Gaddis, T. (2012). *Starting out with programming logic & design* (Custom 2nd ed.). Boston, MA: Pearson Custom	■		
Gaddis, T. (2012). Starting out with programming logic &	■		

Textbook Package	New to this Course	Carried over from Previous Course(s)	Required for Subsequent Course(s)
design lab manual (Custom 3rd ed.). Boston, MA: Pearson Custom.			
Gaddis, T. (2012). *Lab demo media and startup files for starting out with programming logic &design 3E lab manual* CD (Custom 3rd ed.). Boston, MA: Pearson Custom.	■		
Other Items	**New to this Course**	**Carried over from Previous Course(s)**	**Required for Subsequent Course(s)**
Software applications: • Microsoft Visual Studio 2005 or later • MS Visio 2003 or later • MS Office 2003 or later • Raptor 4	■		

Technology Requirements

Minimum Requirements for Computer:

- Pentium III (min.) or equivalent processor
- 256 MB RAM (512 MB preferred)
- 2 GB free space (5 GB preferred) on master drive
- At least 40 GB hard disk for a typical installation
- DVD-ROM drive

Minimum Requirements for Software:

- Windows XP or later
- Microsoft Office 2003 or later
- Internet Explorer 7.0 or later
- Microsoft Visual Basic 2005 or later
- Microsoft Visio 2003 or later
- Raptor 4.0 or later
- Python compiler and IDE

Minimum Requirements for Internet Service (for online access to this course):

- 56Kbps modem (cable or DSL strongly preferred)

Recommended Resources

Books, Professional Journals

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

  is available on FreeTechBooks.com:

  http://freetechbooks.com/about134.html&highlight=allen+downey (accessed 6/5/11)

  - This book teaches readers to think like a computer scientist using Python.
- Smyth, N. *Visual Basic Essentials.*

  http://freetechbooks.com/visual-basic-essentials-t613.html (accessed 6/5/11)

  - This book is intended to provide everything necessary to begin developing Windows applications in Visual Basic.

ITT Tech Virtual Library (accessed via Student Portal)

Books> Books24x7

- Dawson, M. *Python Programming for the Absolute Beginner*. Boston: Premier Press, 2003.
- Goodliffe, P. *Code Craft: The Practice of Writing Excellent Code*. San Francisco: No Starch Press, 2007.
- Rischpater, R. *Beginning Java ME Platform*. New York: Apress, 2008.
- Wright, P. *Beginning Visual Basic 2005 Express Edition: From Novice to Professional*. New York: Apress, 2006.
  (Note: This book uses VB Console.)

School of Study> School of Information Technology> Recommended Links> General

- Tech Fest
- Whatis.com: The IT-specific Encyclopedia

Other References

- Online CS Modules: Algorithms

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

  Self-paced lessons on understanding and using algorithms
- Online CS Modules: Programming Languages

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

  Self-paced lessons on understanding and using various programming languages
- Console Write « Language Basics « VB .NET Tutorial

  http://www.java2s.com/Tutorial/VB/0020__Language-Basics/0080__Console-Write.htm

  Web site providing code samples.

Information Search

Use the following keywords to search for additional online resources that may be used for supporting your work on the course assignments:

- Input, process, and output
- Flowcharts and pseudocode
- Variable declaration, naming, and initialization
- Data types
- Modules
- Local variables vs. global variables
- If-then and if-then-else structures
- Case structures
- While, Do-While, and Do-Until loops
- For loops
- Functions
- Pass by value vs. pass by reference
- File input process
- File output process

NOTE: All links are subject to change without prior notice.

Course Plan

Suggested Learning Approach

In this course, you will be studying individually and within a group of your peers. As you work on the course deliverables, you are encouraged to share ideas with your peers and instructor, work collaboratively on projects and team assignments, raise critical questions, and provide constructive feedback.

Use the following advice to receive maximum learning benefits from your participation in this course:

DO	DON'T
▪ Do take a proactive learning approach ▪ Do share your thoughts on critical issues and potential problem solutions ▪ Do plan your course work in advance ▪ Do explore a variety of learning resources in addition to the textbook ▪ Do offer relevant examples from your experience ▪ Do make an effort to understand different points of view ▪ Do connect concepts explored in this course to real-life professional situations and your own experiences	▪ Don't assume there is only one correct answer to a question ▪ Don't be afraid to share your perspective on the issues analyzed in the course ▪ Don't be negative about the points of view that are different from yours ▪ Don't underestimate the impact of collaboration on your learning ▪ Don't limit your course experience to reading the textbook ▪ Don't postpone your work on the course deliverables – work on small assignment components every day

Course Outline

Unit 1: FUNDAMENTAL CONCEPTS

Unit Duration:
Onsite: 1 week
Online: NA

Upon completion of this unit, the students are expected to:

- Describe the role of software for computers.
- Identify the hardware associated with a computer.
- Describe how computers store data.
- Explain how programs work.
- Differentiate among machine language, assembly language, and high-level languages.
- Differentiate between compilers and interpreters.
- Identify the different types of software.
- Determine program input, processing, and output stages.
- Create the necessary flowcharts to describe a program's structure.
- Use pseudocode to define a program's structure.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
Gaddis, Tony, Chapter 1, pp. 1-22	Assignments	Unit 1 Assignment 1: Homework	1%

	Research Assignments	Unit 1 Research Assignment 1: Exploring Programming Languages	1%
	Labs	Unit 1 Lab 1.1: GUI vs. Console Programming	3%
		Unit 1 Lab 1.2: Using Visual Basic IDE	
		Unit 1 Lab 1.3: Design Tools	

Unit 2: SOFTWARE PROGRAM DESIGN I

Unit Duration:
Onsite: 1 week
Online: NA

Upon completion of this unit, the students are expected to:

- Determine program input, processing, and output stages.
- Create the necessary flowcharts to describe a program's structure.
- Use pseudocode to define a program's structure.
- Formulate solution algorithms for calculations by properly following the order of operations.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
Gaddis, Tony, Chapter 2, pp. 29-55	Assignments	Unit 2 Assignment 1: Homework	1%
	Research Assignments	Unit 2 Research Assignment 1: Researching Variable Naming Rules	1%
	Labs	Unit 2 Lab 2.1: Pseudocode	3%
		Unit 2 Lab 2.2: Flowchart	
		Unit 2 Lab 2.3: Visual Basic	

Unit 3: SOFTWARE PROGRAM DESIGN II

Unit Duration:
Onsite: 1 week
Online: NA

Upon completion of this unit, the students are expected to:

- Create the necessary flowcharts to describe a program's structure.
- Use pseudocode to define a program's structure.
- Describe the use of variables (declaration, naming, assignment, and initialization) in program designs.
- Use the correct data type for variables in program designs.
- Describe the usefulness of properly commenting code.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
Gaddis, Tony, Chapter 2, pp. 56-68	Assignments	Unit 3 Assignment 1: Homework	1%
	Research Assignments	Unit 3 Research Assignment 1: Exploring Reverse Engineering	1%
	Labs	Unit 3 Lab 3.1: Pseudocode	3%
		Unit 3 Lab 3.2: Flowchart	
		Unit 3 Lab 3.3: Visual Basic	
		Unit 3 Lab 3.4: Programming Challenge – Network Systems Administration	

Unit 4: PROGRAM MODULES

Upon completion of this unit, the students are expected to:

- Explain the importance of separating code into modules for efficiency.
- Use flowcharts and pseudocode to represent program modules.
- Describe the impact of program modules on variable scope.
- Describe the necessity of having compatible arguments in module parameters.
- Evaluate the various program modules.

Unit Duration:
Onsite: 1 week
Online: NA

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
Gaddis, Tony, Chapter 3, pp. 75-109	Assignments	Unit 4 Assignment 1: Homework	1%
	Research Assignments	Unit 4 Research Assignment 1: Using Global Variables	1%
	Labs	Unit 4 Lab 4.1: Pseudocode and Modules	3%
		Unit 4 Lab 4.2: Flowchart and Modules	
		Unit 4 Lab 4.3: Visual Basic and Modules	
		Unit 4 Lab 4.4: Challenge: Ping and Website Launches	

Unit 5: DECISIONS I

Upon completion of this unit, the students are expected to:

- Use flowcharts and pseudocode to represent Boolean conditions.
- Apply the concept of nesting conditions to computer programs.
- Use if-then, if-then-else, and case structures in a computer program.
- Compare strings using the program language.
- Use Boolean variables and logical operators in computer programs.
- Use compound logical conditions.

Unit Duration:
Onsite: 1 week
Online: NA

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
Gaddis, Tony, Chapter 4, sections 4.1-4.4, pp. 115-142	Assignments	Unit 5 Assignment 1: Homework	1%
	Research Assignments	Unit 5 Research Assignment 1: Exploring Why and When to Use Shell Scripts	1%
	Labs	Unit 5 Lab 5.1: Evaluating Conditions with Relational Operators	3%
		Unit 5 Lab 5.2: Evaluating Conditions with Logical Operators	
		Unit 5 Lab 5.3: Pseudocode	
		Unit 5 Lab 5.4: Flowcharts	
		Unit 5 Lab 5.5: Visual Basic	

Unit 6: DECISIONS II

Upon completion of this unit, the students are expected to:

- Use flowcharts and pseudocode to represent Boolean conditions.
- Use if-then, if-then-else, and case structures in a computer program.
- Use Boolean variables and logical operators in computer programs.

Unit Duration:
Onsite: 1 week
Online: NA

- Use compound logical conditions.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
Gaddis, Tony, Chapter 4, pp. 142-155	Assignments	Unit 6 Assignment 1: Homework	1%
	Research Assignments	Unit 6 Research Assignment 1: Advantages and Disadvantages of Using Case Structures	1%
	Labs	Unit 6 Lab 6.1: Pseudocode	3%
		Unit 6 Lab 6.2: Flowcharts	
		Unit 6 Lab 6.3: Visual Basic Programming Challenge	
	Exam I	Exam I	25%

Unit 7: REPETITIVE PROCESSING I

Unit Duration:
Onsite: 1 week
Online: NA

Upon completion of this unit, the students are expected to:

- Use pseudocode/flowcharts to represent repetition structures.
- Create the while, do-while, and do-until conditional loops.
- Describe the implications of an infinite loop.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	**Activity/Deliverable Title**	**Grade Allocation** (% of all graded work)
Gaddis, Tony, Chapter 5, pp. 163-183 and pp. 196-201	Assignments	Unit 7 Assignment 1: Homework	1%
	Research Assignments	Unit 7 Research Assignment 1: Exploring How to Stop or Exit an Infinite Loop	1%
	Labs	Unit 7 Lab 7.1: Condition-Controlled with While and Do-While Loops: Pseudocode	3%
		Unit 7 Lab 7.2: Condition-Controlled with While and Do-While Loops: Flowcharts	
		Unit 7 Lab 7.3: Count Controlled with While and Do-While Loops: Pseudocode	
		Unit 7 Lab 7.4: Count Controlled with While and Do-While Loops: Flowcharts	
		Unit 7 Lab 7.5: While and Do While Loops: Visual Basic Challenge I †	
		Unit 7 Lab 7.6: While and Do While Loops: Visual Basic Challenge II	

Unit 8: REPETITIVE PROCESSING I

Unit Duration:
Onsite: 1 week
Online: NA

Upon completion of this unit, the students are expected to:

- Use pseudocode/flowcharts to represent repetition structures.
- Evaluate the counter-controlled For loops.
- Use sentinel values in creating computer programs.
- Use nested loops in a program.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES

	Grading Category	Activity/Deliverable Title	Grade Allocation (% of all graded work)
Gaddis, Tony, Chapter 5, pp. 183-195 and pp. 201-211	Assignments	Unit 8 Assignment 1: Homework	1%
	Research Assignments	Unit 8 Research Assignment 1: Exploring the Differences between Using Count-Controlled Loops and While Loops	1%
	Labs	Unit 8 Lab 8.1: For Loop and Accumulation with Pseudocode	3%
		Unit 8 Lab 8.2: For Loop and Accumulation with Flowcharts	
		Unit 8 Lab 8.3: Accumulation and Loops: Visual Basic Challenge	

Unit 9: FUNCTIONS

Unit Duration:
Onsite: 1 week
Online: NA

Upon completion of this unit, the students are expected to:

- Describe the usefulness of functions in computer programs.
- Write functions using pseudocode/flowcharts.
- Differentiate between the various functions created for use in computer programs.
- Write input validation loops and validation functions to check accuracy of input data.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	Activity/Deliverable Title	Grade Allocation (% of all graded work)
Gaddis, Tony, Chapter 6, pp. 217-218 (through Library Functions) and pp. 225-231(through How to Use Functions) Gaddis, Tony, Chapter 7	Assignments	Unit 9 Assignment 1: Homework	1%
	Research Assignments	Unit 9 Research Assignment 1: Exploring How to Call External Programs or Functions from a Program	1%
	Labs	Unit 9 Lab 9.1: Functions in Pseudocode and Visual Basic	3%
		Unit 9 Lab 9.2: Programming Challenge: Functions and Visual Basic	
		Unit 9 Lab 9.3: Input Validation	
		Unit 9 Lab 9.4: Programming Challenge: Cell Phone Minute Calculator	

Unit 10: FILES

Unit Duration:
Onsite: 1 week
Online: NA

Upon completion of this unit, the students are expected to:

- Describe the different types of files and file access methods.
- Write programs to read data from a file.
- Write programs to write data to a file.

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES

	Grading Category	Activity/Deliverable Title	Grade Allocation (% of all graded work)
Gaddis, Tony, Chapter 10, pp. 361-383	Assignments	Unit 10 Assignment 1: Homework	1%
	Research Assignments	Unit 10 Research Assignment 1: Exploring How to Write Data to a Binary or "Read-Only" File	1%
	Labs	Unit 10 Lab 10.1: File Access and Visual Basic	3%
		Unit 10 Lab 10.2: File Access and Nested Loops	

Unit 11: REVIEW AND FINAL EXAM

Unit Duration:
Onsite: 1 week
Online: NA

READING ASSIGNMENT	GRADED ACTIVITIES / DELIVERABLES		
	Grading Category	Activity/Deliverable Title	Grade Allocation (% of all graded work)
	Exam II	Exam II	25%

† A Candidate for the ePortfolio

Evaluation and Grading

Evaluation Criteria

The graded assignments will be evaluated using the following weighted categories:

Category	Weight
Labs	30%
Assignments	10%
Exam I	25%
Exam II	25%
Research Assignments	10%
TOTAL	**100%**

Grade Conversion

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

Grade	Percentage	Credit
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

Academic Integrity

All students must comply with the policies that regulate all forms of academic dishonesty, or academic misconduct, including plagiarism, self-plagiarism, fabrication, deception, cheating, and sabotage. For more information on the academic honesty policies, refer to the Student Handbook and the Course Catalog.

(End of Syllabus)