

# It210 syllabus essay



This course provides students with a basic understanding of programming practices. Concepts covered include flowcharting, pseudocode methodologies, and an understanding of programming practices. Students will learn how these concepts, when properly applied, improve program design.

Policies Faculty and students/learners will be held responsible for understanding and adhering to all policies contained within the following two documents:

- University policies: You must be logged into the student website to view this document.
- Instructor policies: This document is posted in the Course Materials forum.

University policies are subject to change. Be sure to read the policies at the beginning of each class. Policies may be slightly different depending on the modality in which you attend class. If you have recently changed modalities, read the policies governing your current class modality.

Course Materials Venit, S. , & Drake, E. (2011). Prelude to programming: Concepts and design (5th ed. ). Boston, MA: AddisonWesley. All electronic materials are available on the student website. Week One: Introduction to Software Development Details Objectives 1.

1 Describe the importance of using a structured, modular approach when creating program requirements, design, and code.

1. 2 Identify how a computer processes and stores data. Read the course description and objectives. Read the instructor’s biography and post your own. Read Appendix A. Read Ch. 2 of Prelude to Programming. Read Ch. 3 of Prelude to Programming. Due

Points Course Preparation Reading Reading Reading Course Design Guide  
IT/210 Version 5

2 Reading Participation Discussion Questions CheckPoint Input Data and Output Process Read Ch. 7 of Prelude to Programming. Participate in class discussion. Respond to weekly discussion questions. 12/18/11 12/13/11 12/15/11 12/16/11 10 10 Resources: Appendix B, Appendix C Read the following scenario: You want to build a program that will keep track of your CD and DVD collection at home. Use the table in Appendix C to complete this CheckPoint. Identify at least three processes (capabilities) needed to keep track of your collection. Identify the input data required for each process. Identify a logical name for each data output item and type of data output: real number, integer, or text. 15 Post the table as an attachment. Week Two: Software Application Development Details Objectives

2. 1 Identify the purposes and definitions of software development concepts.

2. 2 Use pseudocode to design a program to solve a computational problem.

2. 3 Identify the application-level requirements of a conversion project.

Match the software development activity or concept with the description or purpose of the activity using the table in Appendix D. Post the table as an attachment. Due Points CheckPoint Software Development Activities and

Purposes CheckPoint Chapter 2 Programming Problem Individual Application-Level Requirements Review the example in Appendix E and the additional examples on pp. 80 & 83 of Prelude to Programming. Complete Programming Problem 2 on p. 109 of Ch. 2 of Prelude to Programming. Resources:

Appendix B and Appendix F Complete the following assignment using

Appendix F: 01/06/12 30 01/08/12 90 Course Design Guide IT/210 Version 5

3 List the application-level requirements for the Currency Conversion project.

Use a structured programming approach to generate an input-process-output chart for the application. Generate the hierarchy chart for the application.

Post the table as an attachment. Week Three: Structured Programming, Part I Details Objectives

3. 1 Demonstrate the sequential processing control structure.

3. 2 Demonstrate the selection processing control structure. Read Ch. 4 of Prelude to Programming. Participate in class discussion. Respond to weekly discussion questions. 01/15/12 01/10/12 01/12/12 01/13/12 10 10 Due Points Reading Participation Discussion Questions CheckPoint Sequential and Selection Processing Control Structure

Read the following scenario: You are an accountant setting up a payroll system for a small firm. Each line of the table in Appendix G indicates an employee's salary range and corresponding base tax amount and tax percentage. Given a salary amount, the tax is calculated by adding the base tax for that salary range and the product of percentage of excess and the amount of salary over the minimum salary for that range. Design a program

that solves this problem. Generate a set of input test values. Perform a design walkthrough to verify your design.

#### 30 Week Four: Structured Programming, Part II Details Objectives

4. Demonstrate the iteration control structure.

4. 2 Design complex program algorithms using the three basic control structures. Read Ch. 5 of Prelude to Programming. Design a program that models the worm's behavior in the following 01/20/12 30 Due Points Reading CheckPoint Course Design Guide IT/210 Version 5 4 Iteration Control Structure scenario: A worm is moving toward an apple. Each time it moves, the worm cuts the distance between itself and the apple by its own body length until the worm is close enough to enter the apple. The worm can enter the apple when it is within one body length of the apple.

Individual Currency Conversion Design Resources: Appendix H and Appendix I Complete the hierarchy chart in Appendix H and the flowcharts in Appendix I, based on the Currency Conversion program requirements and the input-process-output table you generated in Week Two. Develop the pseudocode for the program design. Post the assignment as an attachment. 01/22/12 90

#### Week Five: Complex Data Structures Details Objectives

5. 1 Explain the need for composite complex data structures.

5. 2 Generate the program design and pseudocode for a simple array. Read Ch. 6 of Prelude to Programming. Read Ch. of Prelude to Programming.

Participate in class discussion. Respond to weekly discussion questions.

01/29/12 01/24/12 01/26/12 01/27/12 10 10 Due Points Reading Reading

<https://assignbuster.com/it210-syllabus-essay/>

## Participation Discussion Questions CheckPoint Simple Array Process

Complete Problem 4 on p. 350 in Ch. 6 of Prelude to Programming. You are required to generate only the pseudocode, as described in the Week Two CheckPoint. No charting is required, but you may have to incorporate the bubble sort algorithm on pp. 304-309 to determine the number of salaries above and below the mean. 30 Week Six: Verification and Validation Details

Objectives 6. 1 Recognize how requirements and desk review design are used to verify algorithms.

6. 2 Create test data to validate that algorithms handle user input Due Points Course Design Guide IT/210 Version 5 5 data correctly. CheckPoint Algorithm Verification Answer the following about the information in Appendix J: What will be printed if the input is 0? What will be printed if the input is 100? What will be printed if the input is 51? What will be printed if the user enters Wingding? Is this design robust? If so, explain why. If not, explain what you can do to make it robust.

How many levels of nesting are there in this design? Provide a set of values that will test the normal operation of this program segment. Defend your choices. Provide a set of test values that will cause each branch to be executed. Provide a set of test values that test the abnormal operation of this program segment. 02/03/12 30 Individual Currency Conversion Test Procedure Generate a set of test inputs and expected results for the Currency Conversion program. Post the test procedure as an attachment. 02/05/12 90 Week Seven: File and Database Processing Details

Objectives 7. Determine when a sequential file is more useful than a database.

7. 2 Differentiate between a flat file and a relational database.

7. 3 Design a suitable program to solve given programming problems using the top-down modular approach and pseudocode. Participate in class discussion. Respond to weekly discussion questions. Due Points Participation Discussion Questions CheckPoint Chapter 8 Programming Problems Individual Peer Reviews of Currency Conversion Test Procedure Complete parts a and b of Programming Problems 1 on p. 444 in Ch. 8 of Prelude to Programming.

Provide the analysis and pseudocode only. No diagrams are required.

Perform peer reviews of two classmates' Currency Conversion Test Procedure assignments, which your facilitator will place in your Individual forum on Day 1. Complete the Appendix K form for each peer review. 30 02/08/12 20 Course Design Guide IT/210 Version 5 6 Post the completed Appendix K forms as an attachment.

Week Eight: Object-Oriented Design and Programming Details Objectives

8. 1 Identify both the top-level objects and the GUI interfaces of an electronic product.

8. 2 Describe object-oriented, event-driven programming.

8. 3 Describe a simple object-oriented program.

8. 4 Recognize the difference between object-oriented and structured program design. Read Ch. 9 of Prelude to Programming. Understanding object-oriented methodologies is often difficult. You already understand that object-oriented analysis and design emulates the way human beings tend to think and conceptualize problems. As an example, consider a typical house in which there are several bedrooms, a kitchen, and a laundry room—each with a distinct function. You sleep in the bedroom, you wash clothes in the laundry room, and you cook in the kitchen.

Each room encapsulates all the items needed to complete necessary tasks. You do not have an oven in the laundry room or a washing machine in the kitchen. However, when you do the laundry, you do not just add clothes to the washer and wait in the laundry room; once the machine has started, you may go into the kitchen and start cooking dinner. How do you know when to go back to check the laundry? When the washer buzzer sounds, a message is sent to alert you to go back into the laundry room to put in a new load. While you are folding clothes in the laundry room, the oven timer may ring to inform you that your dinner is done cooking.

What you have is a set of well-defined components: Each provides a single service to communicate with the other components using simple messages when something needs to be done. If you consider a kitchen, you see it is composed of several, smaller components, including the oven, refrigerator, and microwave. Toplevel objects are composed of smaller components that do the actual work. This perspective is a very natural way of looking at the world, and one with which everyone is familiar. The same thing is done in



object-oriented programming: • • Identify components that perform a distinct service.

Encapsulate all items in the component necessary to get the job done.

02/15/12 30 Due Points Reading CheckPoint Interfaces and Communication Messages Course Design Guide IT/210 Version 5 7 Identify the messages that need to be provided to the other components. Although the details can be quite complex, they are the basic principles of object-oriented programming. Consider the microwave oven in your kitchen, using the objectoriented thinking described above. Create a table with the following four column headings: Top-Level Objects, Communicates With, Incoming Messages, and Outgoing Messages. Identify the top-level objects of the microwave. Explain some graphical user interfaces and communication messages that occur during the operation of a microwave. Describe some of the advantages of having a componentized system. For example, what happens if the microwave breaks? Post your completed CheckPoint as an attachment.

CheckPoint Object-Oriented Data and Processes Identify a task you perform regularly, such as cooking, mowing the lawn, or driving a car. Write a short, structured design—using pseudocode only—that accomplishes this task.

Think about this task in an object-oriented way and identify the objects involved in the task. Identify how you can encapsulate the data and processes you identified into an object-oriented design. Describe the architectural differences between the object-oriented and structured designs. Which of the designs makes more sense to you? Why? Generate an object-oriented design for a system that keeps tracks of your CD and DVD collection. Identify each of the classes, associated data, and operations for

the classes. Generate the pseudocode for each of the classes as demonstrated on p. 458 in Ch. of Prelude to Programming. Draw a GUI that will create the objects and provide access to each object's processing methods. Note. Use the drawing tool in Microsoft® Word or in any other applicable drawing tool to complete this part of the assignment. 02/17/12 30 Individual Object-Oriented Design 02/19/12 90 Week Nine: Requirement Specifications, Design Specifications, and Processing Models Details Due Points Course Design Guide IT/210 Version 5 8 Objectives 1. 1 Create requirement specifications, design specifications, and processing models, including input/out tables, data flow models, and procedural processing.

Participate in class discussion. Respond to the capstone discussion question. Drawing upon your knowledge of software development, which process—requirements, design, coding, or testing—do you think has the greatest effect on the overall success and quality of development? Explain your answer. 02/26/12 250 02/26/12 02/22/12 10 15 Participation Capstone Discussion Question Final Project Currency Conversion Consolidate all sections of the Currency Conversion development documentation: menu selection, requirements, design, and testing. Incorporate any changes recommended by the facilitator.

Post the assignment as an attachment. Copyright University of Phoenix® is a registered trademark of Apollo Group, Inc. in the United States and/or other countries. Microsoft®, Windows®, and Windows NT® are registered trademarks of Microsoft Corporation in the United States and/or other countries. All other company and product names are trademarks or registered trademarks of their respective companies. Use of these marks is <https://assignbuster.com/it210-syllabus-essay/>

not intended to imply endorsement, sponsorship, or affiliation. Edited in accordance with University of Phoenix® editorial standards and practices.