

OUTCOMES BASED LEARNILNG MATRIX

Course: CTIM281 Software Design and Development (3 credits, 60 hours)

Department: Computer Technology and Information Management

Description:

This course presents the fundamentals of developing programming logic. It utilizes a language-independent approach to programming. Universal programming concepts are presented to encourage logical thinking to take a problem from development to a strong working solution. A variety of tools are used to prepare students for programming situations. Topics include sequence, selection, and repetition with an introduction to object-orientated concepts. Two lecture and two laboratory hours per week.

Prerequisite: None

While completing the table below, remember that the individual outcomes you list in the first column should answer this question: **What must the learner know and be able to do at the end of the course?** Items in the third column should answer the question: **How do we know?** The second column is where teachers can be most creative; it's for pedagogy. Each rectangle in column one should contain just one outcome; the corresponding rectangles in columns two and three, however, may contain more than one item. Using the code at the end of the matrix, indicate the core competencies being strengthened by the outcomes activities and the assessment tools.

*COURSE OUTCOMES	OUTCOMES ACTIVITIES	ASSESSMENT TOOLS
At the end of this course students will have a basic understanding of developing logic to code programs. 1. Have basic understanding of a computer system, simple program	1. Instructor lead tasks: a. explain the purpose of flowcharts (CCT, IL, IL WC) b. explain the purpose of pseudocode	1. In Class: Perform Maintenance Find the Bug Flowchart/Pseudocode Up for Discussion (CCT, IL, QL, IL WC)

Approved by CTIM Department—September 2015

<p>logic, and the tools to document basic logic.</p>	<p>(CCT, IL, IL WC) c. perform maintenance to existing code (CCT, IL, OC, IL WC) d. perform find the bug to existing code (CCT, IL, OC, IL WC) e. develop new flowchart/pseudocode (CCT, IL, OC, IL WC)</p>	<p>2. Homework: Flowchart/Pseudocode Assignment (CCT, IL, QL, IL WC) 3. Chapter Review Quiz: Objective test. (IL WC)</p>
<p>2. Be able to demonstrate elements of high-quality programs and apply that to declaring variables, performing arithmetic operations, modularization. and common configuration for Mainline logic.</p>	<p>2. Instructor lead tasks: a. explain good program design (CCT, IL, IL WC) b. explain: hierarchy and modularization (CCT, IL, IL WC) c. perform maintenance to existing code (CCT, IL, OC, IL WC) d. perform find the bug to existing code (CCT, IL, OC, IL WC) e. develop new flowchart/pseudocode (CCT, IL, OC, IL WC)</p>	<p>1. In Class: Perform Maintenance Find the Bug Flowchart/Pseudocode Up for Discussion (CCT, IL, QL, IL WC) 2. Homework: Flowchart/Pseudocode Assignment (CCT, IL, QL, IL WC) 3. Chapter Review Quiz: Objective test. (IL WC)</p>
<p>3. Be able to demonstrate elements of high-quality programs and apply that to declaring variables, performing arithmetic</p>	<p>2. Instructor lead tasks: a. explain good program design (CCT, IL, IL WC) b. explain: hierarchy and</p>	<p>1. In Class: Perform Maintenance Find the Bug Flowchart/Pseudocode</p>

<p>operations, modularization. and common configuration for Mainline logic.</p>	<p>modularization (CCT, IL, IL WC) c. perform maintenance to existing code (CCT, IL, OC, IL WC) d. perform find the bug to existing code (CCT, IL, OC, IL WC) e. develop new flowchart/pseudocode (CCT, IL, OC, IL WC)</p>	<p>Up for Discussion (CCT, IL, QL, IL WC)</p> <p>2. Homework: Flowchart/Pseudocode Assignment (CCT, IL, QL, IL WC)</p> <p>3. Chapter Review Quiz: Objective test. (IL WC)</p>
<p>4. Be able to demonstrate the 3 basic structures; sequence, selection, and loop.</p>	<p>4. Instructor lead tasks: a. explain selection (CCT, IL, IL WC) b. explain selection; Boolean and relational operators (CCT, IL, IL WC) c. perform maintenance to existing code (CCT, IL, OC, IL WC) d. perform find the bug to existing code (CCT, IL, OC, IL WC) e. develop new flowchart/pseudocode (CCT, IL, OC, IL WC)</p>	<p>1. In Class: Perform Maintenance Find the Bug Flowchart/Pseudocode Up for Discussion (CCT, IL, QL, IL WC)</p> <p>2. Homework: Flowchart/Pseudocode Assignment (CCT, IL, QL, IL WC)</p> <p>3. Chapter Review Quiz: Objective test. (IL WC)</p>

<p>5. Be able to demonstrate loops. Have an understanding of accumulators.</p>	<p>5. Instructor lead tasks:</p> <ul style="list-style-type: none"> a. explain the purpose of loops (CCT, IL, IL WC) b. explain nested loops (CCT, IL, IL WC) c. perform maintenance to existing code (CCT, IL, OC, IL WC) d. perform find the bug to existing code (CCT, IL, OC, IL WC) e. develop new flowchart/pseudocode (CCT, IL, OC, IL WC) 	<ul style="list-style-type: none"> 1. In Class: <ul style="list-style-type: none"> Perform Maintenance Find the Bug Flowchart/Pseudocode Up for Discussion (CCT, IL, QL, IL WC) 2. Homework: <ul style="list-style-type: none"> Flowchart/Pseudocode Assignment (CCT, IL, QL, IL WC) 3. Chapter Review Quiz: Objective test. (IL WC)
<p>6. Be able to demonstrate storing and searching arrays with loops</p>	<p>6. Instructor lead tasks:</p> <ul style="list-style-type: none"> a. explain storing arrays (CCT, IL, IL WC) b. explain searching arrays (CCT, IL, IL WC) c. perform maintenance to existing code (CCT, IL, OC, IL WC) d. perform find the bug to existing code (CCT, IL, OC, IL WC) e. develop new flowchart/pseudocode 	<ul style="list-style-type: none"> 1. In Class: <ul style="list-style-type: none"> Perform Maintenance Find the Bug Flowchart/Pseudocode Up for Discussion (CCT, IL, QL, IL WC) 2. Homework: <ul style="list-style-type: none"> Flowchart/Pseudocode Assignment (CCT, IL, QL, IL WC) 3. Chapter Review Quiz: Objective

	(CCT, IL, OC, IL WC)	test. (IL WC)
7. Have an understanding of method design issues, including implementation hiding, cohesion, and coupling	<p>7. Instructor lead tasks:</p> <ul style="list-style-type: none"> a. explain the methods (CCT, IL, IL WC) b. perform maintenance to existing code (CCT, IL, OC, IL WC) c. perform find the bug to existing code (CCT, IL, OC, IL WC) d. develop new flowchart/pseudocode (CCT, IL, OC, IL WC) 	<ul style="list-style-type: none"> 1. In Class: <ul style="list-style-type: none"> Perform Maintenance Find the Bug Flowchart/Pseudocode Up for Discussion (CCT, IL, QL, IL WC) 2. Homework: <ul style="list-style-type: none"> Flowchart/Pseudocode Assignment (CCT, IL, QL, IL WC) 3. Chapter Review Quiz: Objective test. (IL WC)
8. Have an understanding of Object-Oriented Programming and more advanced OOP concepts.	<p>8. Instructor lead tasks:</p> <ul style="list-style-type: none"> a. explain objected oriented programming (CCT, IL, IL WC) b. perform maintenance to existing code (CCT, IL, OC, IL WC) c. perform find the bug to existing code (CCT, IL, OC, IL WC) d. develop new flowchart/pseudocode 	<ul style="list-style-type: none"> 1. In Class: <ul style="list-style-type: none"> Perform Maintenance Find the Bug Flowchart/Pseudocode Up for Discussion (CCT, IL, QL, IL WC) 2. Homework: <ul style="list-style-type: none"> Flowchart/Pseudocode Assignment (CCT, IL, QL, IL WC)

	(CCT, IL, OC, IL WC)	3. Chapter Review Quiz: Objective test. (IL WC)
To strengthen Core Competencies** in order to increase success in this and other courses and in the workplace.	Referenced above	Referenced above.

*Try to express an outcome as an infinitive phrase that concludes this sentence: **At the end of the course, the students should be able to . . .** Finding the line between too general and too specific can be difficult. In an English Composition course, for instance, it is probably too general to say, "The student should be able to write effective essays." It is probably too specific to say, "The student should be able to write an introductory paragraph of at least 50 words, containing an attention-getting device, an announcement of the narrowed topic, and an explicit thesis sentence." Just right might read, "The student will write introductions that gather attention and focus the essay."

**Indicate the Core Competencies that apply to the outcomes activities and assessment tools: critical and creative thinking (CCT); oral communications (OC); quantitative literacy (QL); information literacy (IL); written communication (WC); civic engagement (CE); integrative learning (IG); global learning (GL).