

OUTCOMES BASED LEARNING MATRIX

Course: CTIM 285 Python

(3 credit, 4 contact hours)

Department: Computer Technology and Information Management

Description:

This course will give the student an understanding of Python, an object-oriented scripting language, and the role of Python in the object-oriented and scripting worlds. The student will create simple programs using sequence, selection, repetition and functions and develop advanced Python programming skills that are required to fully utilize the capabilities of this object-oriented scripting language; objects, classes, strings, lists, inheritance, polymorphism, files, tuples/sets/dictionaries, exceptions, recursion, and GUI programming. The student will create sophisticated programs.

Prerequisite: Introduction to Java Programming (CTIM157) or Programming in C++ (CTIM371) or Departmental Approval.

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. Use 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 be able to create simple programs using Python to include.		

<p>1. Explore the Python programming language</p>	<ol style="list-style-type: none"> 1. Explore basic computer architecture (CCT, OC, QL, IL, WC, IG) 2. Explore basic Internet architecture (CCT, OC, QL, IL, WC, IG) 3. Design Python programs (CCT, OC, QL, IL, WC, IG) 4. Compile Python programs (CCT, OC, QL, IL, WC, IG) 5. Run Python programs (CCT, OC, QL, IL, WC, IG) 	<p>1. Quizzes, tests, projects, class participation, homework assignments (CCT, OC, QL, IL, WC, IG).</p>
<p>2. Work with types: numbers, strings, lists, dictionaries, tuples, and files</p>	<ol style="list-style-type: none"> 3. Use/manipulate types (CCT, OC, QL, IL, WC, IG) 4. Use/manipulate numbers (CCT, OC, QL, IL, WC, IG) 5. Use/manipulate strings (CCT, OC, QL, IL, WC, IG) 6. Use/manipulate lists (CCT, OC, QL, IL, WC, IG) 7. Use/manipulate dictionaries (CCT, OC, QL, IL, WC, IG) 8. Use/manipulate tuples (CCT, OC, QL, IL, WC, IG) 9. Use/manipulate files (CCT, OC, QL, IL, WC, IG) 	

<p>3. Manipulate the flow of control</p>	<ol style="list-style-type: none"> 1. Create/use assignment statements, expression statements and print operations (CCT, OC, QL, IL, WC, IG) 2. Create/use if tests (CCT, OC, QL, IL, WC, IG) 3. Explore syntax rules (CCT, OC, QL, IL, WC, IG) 4. Create/use while and for loops types (CCT, OC, QL, IL, WC, IG) 5. Create/use iterations and list comprehensions (CCT, OC, QL, IL, WC, IG) 6. Explore documentation (CCT, OC, QL, IL, WC, IG) 	
<p>4. Design/create/use functions</p>	<ol style="list-style-type: none"> 1. Explore function basics (CCT, OC, QL, IL, WC, IG) 2. Examine scope (CCT, OC, QL, IL, WC, IG) 3. Create/use arguments (CCT, OC, QL, IL, WC, IG) 4. Create/use iterations and list comprehensions (CT, TS, R, W) 	
<p>1. Work with modules</p>	<ol style="list-style-type: none"> 1. Examine module structure (CCT, OC, QL, IL, WC, IG) 2. Use/manipulate module coding (CCT, OC, QL, IL, WC, IG) 3. Use/manipulate module packages (CCT, OC, QL, IL, WC, IG) 	

2.Create/use classes	1.Examine class structure (CCT, OC, QL, IL, WC, IG) 3.Examine basic class coding (CCT, OC, QL, IL, WC, IG) 4.Examine advanced class coding (CCT, OC, QL, IL, WC, IG) 5.Examine operator overloading (CCT, OC, QL, IL, WC, IG) 6.Explore designing classes (CCT, OC, QL, IL, WC, IG)	
3.Program exception handling and tools	1.Examine exception basics (CCT, OC, QL, IL, WC, IG) 2.Examine exception coding (CCT, OC, QL, IL, WC, IG) 3.Explore exception objects (CCT, OC, QL, IL, WC, IG) 4. Explore designing with exceptions (CCT, OC, QL, IL, WC, IG)	
To strengthen core competencies** in order to increase success in this and other courses and in the workplace.		

*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).