



Capacity Building in Higher Education

INNOVATIVE ICT EDUCATION FOR SOCIAL-ECONOMIC DEVELOPMENT (IESED)



COURSE AND SYLLABUS DESIGN

2018

Coordinating institution
ALYTUS KOLEGIJA University of Applied Sciences

Contents

1. Course design	3
1.1. Who are the students?	3
1.2. What do I want students to be able to do?	3
1.3. How will I measure students' abilities?	4
2. Didactic Relation Model.....	4
2.1. Main elements of learning course	4
2.2. Course Teaching Methods	5
2.3. Course Evaluation Methods.....	5
3. Requirements for learning content	6
4. An example of course "Introduction to Programming Using Python".....	6
4.1 Syllabus and Schedule	6
4.1.1. Course description	6
4.1.2. Learning Outcomes	6
4.1.3. Grading Policy	7
4.1.4. Course Schedule	7
Bibliography	10

1. Course design

Effective course design begins with understanding who your students are, deciding what you want them to learn; determining how you will measure student learning; and planning activities, assignments and materials that support student learning. For all interactions with students plan ahead by ask yourself:

1. Who are the students?
2. What do I want students to be able to do?
3. How will I measure students' abilities?

By asking yourself these questions at the onset of your course design process you will be able to focus more concretely on learning outcomes, which has proven to increase student learning substantially as opposed to merely shoehorning large quantities of content into a quarters worth of class meetings.

1.1. Who are the students?

Before the class begins, find out as much as you can about the students. Consider the level of your course and the type and level of student that typically enrolls in this course. If you are new to teaching the course you may want to consult with colleagues who have previously taught the course to gather some of this information. It is also helpful to review your class roster before the quarter starts.

- Are your students new to the university? Are they new to the topic of the course or the department?
- What are students' motivations for taking the course?
- What might you expect students to know before the first class? Consider previous courses they may or may not have taken. Are the students majors in your department or are they fulfilling a distribution credit?
- What range of backgrounds and previous experience is typically represented among students in this class?
- What problems do students typically have with this material at this level?

1.2. What do I want students to be able to do?

Once you have considered who the students in your course are, ask yourself what they should be able to do at the end of the course. Try to answer this question as specifically as you can by using terms that emphasize student abilities you can measure or easily recognize. For example, it can be more challenging to measure students abilities based on what they may know or understand as opposed to measuring their abilities to preform tasks such as identify, differentiate, apply or produce. This process will help you solidify your **course goals**.

Tools that can help you design course objectives:

- Bloom's Taxonomy of Educational Objectives

1.3. How will I measure students' abilities?

Designing your course around activities that are most likely to lead students towards the goals you have defined will help them acquire and retain skills longer. Some goals can be achieved through listening to lecture or reading assigned texts. Others may require more active experimentation, practice or discussion. For example, writing, discussions, field work, service learning, problem solving or small group collaboration. No matter what combination of activities you choose always keep in mind how the core activity, as opposed to subject content, will progress students' abilities.

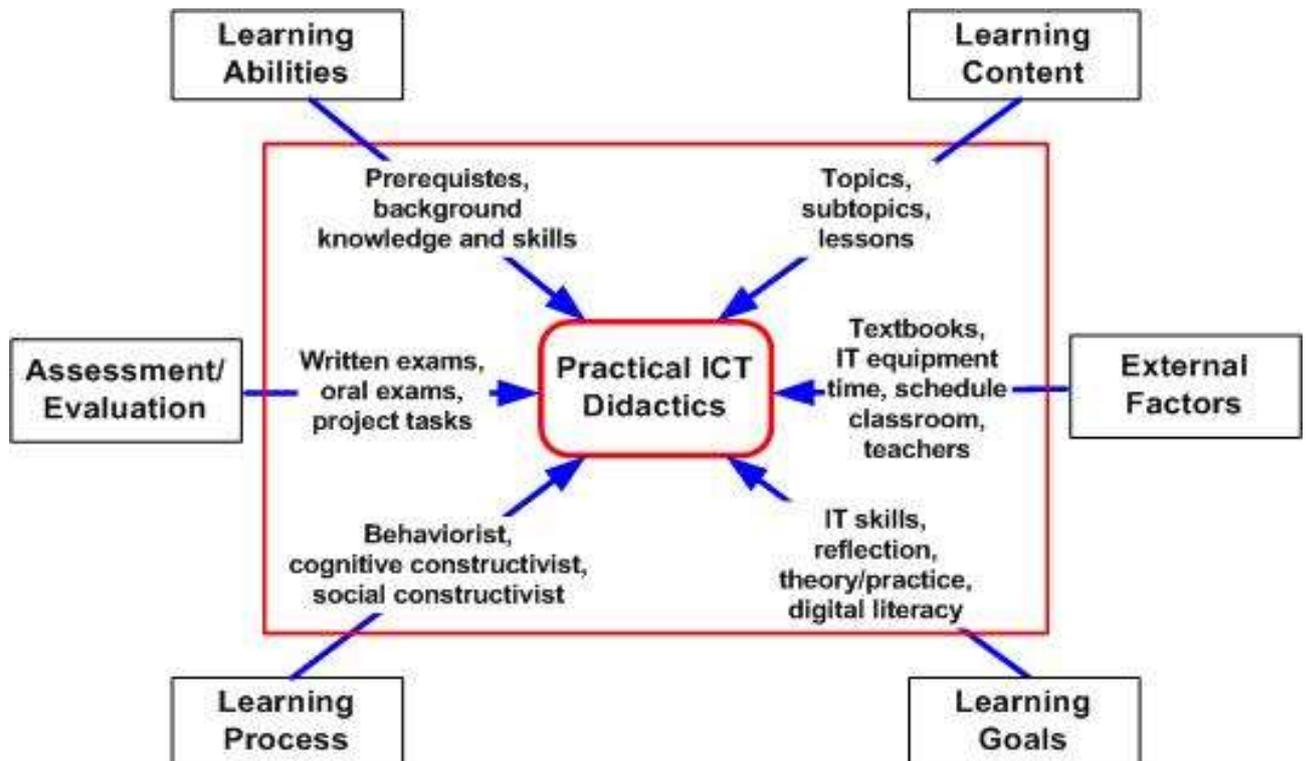
What will provide you with reliable evidence *during the course* that your students are learning and *at the end of the course* that they have obtained/mastered the abilities you envisioned at the beginning of the course? This is the part where you choose assignments, activities and other methods of assessment. For example, will you have weekly quizzes? objective tests? original research papers? presentations? performances? group or individual projects? Assessment is an important aspect of student learning. Make sure to think carefully when pairing assessments with learning objectives. For more on assessment design see our [Assessing student learning](#) page.

2. Didactic Relation Model

2.1. Main elements of learning course

Assumes that different parts of the educational system are related to each other, and that there is a reciprocal influence between the elements:

- Students' characteristics and prerequisite knowledge
- Course content
- Learning goals
- Methods of working and teaching methods
- Assessment procedures
- Learning environment, conditions and resources
- Students' abilities are prerequisite knowledge and skills, educational background and experience, as well as personal experiences.
- External conditions are factors that make learning possible, such as computer equipment, resources, library, books, place, classroom settings, economical conditions, legal and ethical conventions, curriculum, time table, syllabus, etc.
- Learning goals are about what the students should possess after finishing the training course in terms of concepts, methods, theory, practices, ideas and principles.
- Learning content is learning material that is associated with the subject matter, its topics and subtopics and how these are broken down into lectures.
- Learning process is concerned with methods and activities based on learning theories, such as reading textbooks, doing exercises, performing projects, as well as the process of changing students' knowledge to new knowledge.
- Assessment is the process of assessing the student learning and how this can be done, such as oral and written exams, writing a report, performing a project, etc.



2.2. Course Teaching Methods

- Lectures
- Group work
- Project work
- Practical exercises
- Discussion forum
- Research work
- Presentations
- Quizzes
- Oral and written exams
- Intensive weekend seminars may be part of the course...

2.3. Course Evaluation Methods

- Two standard questionnaires (pre- and post questionnaires)
- Classroom dialog and discussions
- Observations and discussions during the supervision of compulsory and project work
- Oral and written exams
- Research work and reports
- Assignments
- Quizzes....

3. Requirements for learning content

1. Theoretical part:
 - a. Minimum 40 pages of theoretical part for 3 credits subject (6 credits – minimum 80 pages)
 - b. Chapters must be short, clearly readable. It should be broken down into short lectures, subtopics.
2. Practical part:
 - a. Assignments
 - b. Quizzes
 - c. Midterm Exam.
 - d. Final Exam...

4. An example of course “Introduction to Programming Using Python“

This example of course **Introduction to Programming Using Python** you can find in website <https://www.edx.org/>. In this website are many **learning courses of best universities of the world for free.**

4.1 Syllabus and Schedule

4.1.1. Course description

- This nine week course is an introduction to Python programming language. The course is open to all learners who hope to gain an understanding of the basic components of computer programming. No previous knowledge of programming is required.
- You will learn basic concepts and terminologies such as variables, operators, expressions, conditional statements, loops, and functions. Lessons include videos, notes, solutions, references, tables, and exercises. The exercises in each lesson are based on interactive hands-on questions and programming practices. These exercises are designed to help you understand the components of Python language while incrementally improving your analytical and programming skills. **Exercises are not graded**, they do not have due dates, and they are available at all times.
- There are also graded modules which include 4 assignments, 6 quizzes, a midterm exam, and a final exam. **Graded modules have release dates and due dates.** In order to earn a certificate you must have a score greater than **60%** in the graded modules.

4.1.2. Learning Outcomes

Upon completing this course you will be able to:

- Understand Python expressions and statements
- Get user-input data and manipulate it.
- Understand simple and compound data types such as integers, floats, strings, lists, dictionaries, and tuples.
- Implement structures such as if statements and loops
- Read from and write to external data files using Python.
- Implement functions and call built-in Python functions.

- Import and use library function modules from the Python library.
- Handle programming errors and exceptions.
- Implement useful and fun programs such as searching, sorting, and solving Sudoku and crossword puzzles.

4.1.3. Grading Policy

Module(s)	Percent (Weight)
Assignments (4)	25%
Quizzes (6)	25%
Midterm Exam.	25%
Final Exam.	25%

4.1.4. Course Schedule

Lesson		Unit
Week 1	Python Installation and Environment	Installing Python on Windows
		Installing Python on Mac
		Installing Python on Linux
		Python Shell & IDLE
	Getting Started	Python Basics
		Algorithms and Programs
		Your First Program
		Using the Text Editor
Week 2	Variables and Identifiers	Variables and Identifiers
		How to Name Your Identifiers
		Use Descriptive Identifiers and Comments
		Variable Types
		Get Input from User
	Statements	Equal Sign
		Statements, Assignments, and Expressions
		List Basics
	List Basics Exercise 1	
Quiz 1		
Week 3	Operators	Operators
		Arithmetic Operators and Precedence
		Relational Operators
		Logical Operators
		Membership Operators
	Conditionals	Conditionals if else
		Conditionals if elif else
Quiz 2:		
Week 4	Loops	While Loop
		For Loop

		Continue and Break
		Nested Loops
	Functions	Functions
		Functions Analogy
		Modules
Assignment 1		
Quiz 3		
Week 5	List Manipulation	List Methods
	List slicing	List Slicing
		List Slicing Negative Index
		List Slicing with Step
		List Slicing Default Index
Midterm Exam		
Week 6	Strings	Strings
		Character Encoding
	String Methods	String Methods
		Strings Practice 1
		Strings Practice 2
Assignment 2		
Quiz 4		
Week 7	Using another IDLE	Why Use Another IDLE
		Step by step guide to install Wing IDE 101
	Multi-Dimensional Lists	Multidimensional Lists
	Dictionaries	Dictionaries
		Dictionary Methods
Assignment 3		
Quiz 5		
Week 8	File I/O	File IO
		File Access Modes
		File Position
	Tuples	Tuples
		Tuples in Function Returns
	Formatting	Formatting
		Formatting Positional
		Formatting Width Precision Type
		Formatting Practice
		Formatting Fill Align
		Formatting Sign
Assignment		
Quiz 6		
Week 9	More about Functions	Namespaces
		Scope
		Recursive Functions

	Error Handling	Errors and Exceptions
		Exception Handling
Final Exam.		

4.2. Content of course (titles of lectures):

Installing Python: Windows, Mac, Ubuntu

Python Shell and IDLE

Python Basics

Variables and Identifiers

Statements

List Basics

Operators, Arithmetic and Precedence

Operators, relational, ..

Conditionals

Loops

Nested Loops

Functions

Modules

List Methods

List Slicing

List slicing with Steps

Strings

String Methods

Using another IDE

Multidimensional Lists

Dictionaries

File I/O

Tuples

Formatting

Formatting, Width, ..

Formatting, Fill, ..

Namespaces and Scope

Recursion

Error Handling

Other Tutorials:

[Beginner's Guide to Python](#)

[Python 3.4.4 documentation](#)

[The Python Tutorial \(V 3.4\)](#)

[Learn Python the Hard Way](#)

[Python Tutor](#)

[Codecademy](#)

Bibliography

- Davis, B. G. (1993). Designing or revising a course. In *Tools for teaching* (pp. 3-20). San Francisco, CA: Jossey-Bass.
- O'Brien, J. G., Millis, B. J., & Cohen, M. G. (2008). The course syllabus: A learning-centered approach. San Francisco, CA: Jossey-Bass.
- Prigent, R. (2000). Charting your course: How to prepare to teach more effectively. Madison, WI: Atwood Publishing.
- Svinicki, M. D., & McKeachie, W. J. (2011). Countdown for course preparation. In *McKeachie's teaching tips: Strategies, research, and theory for college and university teachers* (pp. 10-20). Belmont, CA: Wadsworth.
- Wehlburg, C. M. (2006). Meaningful course revision: Enhancing academic engagement using student learning data. Bolton, MA: Anker Publishing Company, Inc.