CSC 481 Final REPORT

CSC101 DataCamp Introductory Python Course

Matthew Hancock & Rachel Burk

# Identify the Client

Dr. Nelson & Computer Science Program for Saint Martin’s University

# Identify the Client’s Needs

Creating an introductory python course based off the material from the book “Python for Everybody” for the university’s CSC101 introductory programming language class.

# Literature Review

We reviewed pedagogical topics specifically from University of Colorado’s Center for Faculty Development. The process for designing this course will require three stages.

The first stage of the course design will be to outline each chapter that we will cover in the DataCamp Light course. The Python for Everybody book already has chapters broken into no more than 16 subsections. The benefit of this structure is that it helps to structure our course as well as influence learning objectives. The first stage’s primary focus is to write “observable learning objectives” which according to the University of Colorado’s Center for Faculty Development states they “provide students with a clear purpose to focus their learning efforts.” A great distinction provided in this material is the importance of learning goals and learning objectives. The difference between them is “learning Goals are what you hope to accomplish in your course: the overall goals that do not necessarily result in products of observable and measurable behavior”. While “learning objectives are brief, clear statements about what students will be able to do when they complete instruction”. This is significant for course design because of how concise learning objectives must be in order to check that students will complete the material successfully.

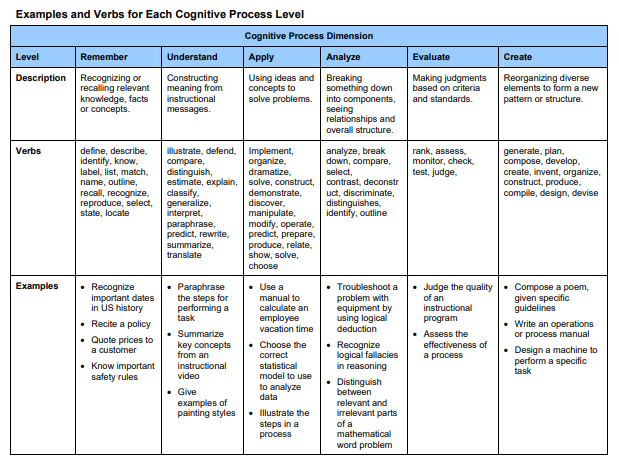
As mentioned earlier, learning objectives need to be observable and measurable. Learning objectives include the following:

* Condition - the condition under which the student will perform the described behavior
* Behavior - a description of a specific, observable behavior
* Degree - the degree indicates the desired level or degree of acceptable performance

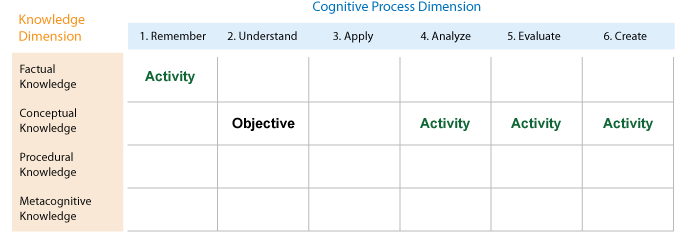
This course for example, a learning objective for the first chapter would be the

* condition: “identify the reserved words in python”
* behavior: students will be able to pick the reserved python words from a list
* degree: students will be able to identify reserved words 80% of the time.

Below is a table of observable behaviors that we will be using in order to construct our learning objectives. Ideally, each module will have multiple learning objectives per chapter and will be tested with a variety of process “levels.”



The second stage of the course design will be to create interactive activities and assessments to guide learning. Activities will depend greatly on the process level decided for the learning objective. DataCamp light offers the ability to utilize multiple choice and console practices as both activities and assessments. DataCamp light uses remembering, understanding, applying, and creating levels. This benefits the student because they will be assessed the same way they are practicing. This will also be the case throughout their learning journey with Saint Martin’s University. Through my experience, professors use similar tools to create activities and assessments. As visualized below, utilizing many process levels to achieve a learning objective may be most effective. Multiple-choice questions are great for an activity that requires remembering and memorizing. Reserved words, for example, would greatly benefit from multiple-choice. The console activities allow students to apply and create code in order to understand more hands-on topics such as loops.



The third stage is implementation and documentation. The implementation will take place through GitHub and DataCamp Light. This will require us to understand the commands that DataCamp Light recognizes and structure our activities in that way. Doing this process with GitHub allows us to document and implement the material. This is especially important because there is a great likelihood that Python for Everybody through DataCamp Light will need to be worked on by a second senior team. The way that we will handle this transition is to comment our content well, and to leave detailed instructions on the direction we were looking to head for the content.

The goal for this project is to complete the first 6 chapters at least. This gives us the opportunity to work with Dr. Nelson on the content as well as fish out any bugs that might arise with the code. Future reviews of literature describing the best methods to create multiple-choice and console content will be required as we continue to develop the course.

# Methodology

## Objective and Motivation

It is our goal to create and comprehensive and useful DataCamp course that will be used to teach students in an accessible and engaging medium. We believe that we could apply our programming skills to benefit future computer science students of Saint Martin’s University. This DataCamp course would most likely be used in CSC101 to introduce programming to future computer science students.

## Learning Objectives

Module 1: Brief introduction to the python course and explanation of additional resources as well as giving credit to the Python for Everybody book and content.

Module 2: Variables, expression and statements. In this module we will be covering the basics of programming languages. With a large focus on values, types, variables and operators. We will also cover the computer uses order of operations to make sense of the code.

Module 3: Conditional Execution. Introduce Boolean Expressions and Logical Operations. We will also cover conditional execution and different kinds of conditionals.

Module 4: Functions. Introduce how to call and declare functions. Discuss and explain the definition of a function and its uses in programming.

Module 5: Iteration. This chapter will have a focus on iteration and looping. We will introduce the basics of for and while loops as well as the dangers of infinite looping.

Module 6: Strings. We will take about the basics of strings and how they are used in programming. We will cover string methods, comparison and the importance of parsing strings.

Module 7: Files. We will cover the basics of writing and reading to files in a directory. We will cover what persistence means as well as how to search through a file and edit it.

# Course Design: Content Modules & Page Numbers

## 1 Introduction to DataCamp & Python

* 1. What is DataCamp?
  2. What is Python?
  3. Python for Everybody Book
  4. Additional Resources & Practice Problems

## 2 Variables, expressions, and statements p.19

2.1 Values and types . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

2.2 Variables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

2.3 Variable names and keywords . . . . . . . . . . . . . . . . . . .21

2.4 Statements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

2.5 Operators and operands . . . . . . . . . . . . . . . . . . . . . . . 22

2.6 Expressions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

2.7 Order of operations . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

2.8 Modulus operator . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

2.9 String operations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

2.10 Asking the user for input . . . . . . . . . . . . . . . . . . . . . . 25

2.11 Comments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

2.12 Choosing mnemonic variable names . . . . . . . . . . . . . 27

## 3 Conditional execution p.31

3.1 Boolean expressions . . . . . . . . . . . . . . . . . . . . . . . . . . 31

3.2 Logical operators . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

3.3 Conditional execution . . . . . . . . . . . . . . . . . . . . . . . . . 32

3.4 Alternative execution . . . . . . . . . . . . . . . . . . . . . . . . . .33

3.5 Chained conditionals . . . . . . . . . . . . . . . . . . . . . . . . . . 34

3.6 Nested conditionals . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

3.7 Catching exceptions using try and except . . . . . . . . . . 36

3.8 Short-circuit evaluation of logical expressions . . . . . . 38

## 4 Functions p.43

4.1 Function calls . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

4.2 Built-in functions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

4.3 Type conversion functions . . . . . . . . . . . . . . . . . . . . . . 44

4.4 Math functions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

4.5 Random numbers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

4.6 Adding new functions . . . . . . . . . . . . . . . . . . . . . . . . . 47

4.7 Definitions and uses . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

4.8 Flow of execution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49

4.9 Parameters and arguments . . . . . . . . . . . . . . . . . . . . . . 49

4.10 Fruitful functions and void functions . . . . . . . . . . . . . 51

4.11 Why functions? . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52

## 5 Iteration p.57

5.1 Updating variables . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

5.2 The while statement . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

5.3 Infinite loops . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58

5.4 Finishing iterations with continue . . . . . . . . . . . . . . . . 59

5.5 Definite loops using for . . . . . . . . . . . . . . . . . . . . . . . . 60

5.6 Loop patterns . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61

5.6.1 Counting and summing loops . . . . . . . . . . . . . . . . . . 61

5.6.2 Maximum and minimum loops . . . . . . . . . . . . . . . . 62

## 6 Strings p.67

6.1 A string is a sequence . . . . . . . . . . . . . . . . . . . . . . . . . . 67

6.2 Getting the length of a string using len . . . . . . . . . . . . . 68

6.3 Traversal through a string with a loop . . . . . . . . . . . . . 68

6.4 String slices . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69

6.5 Strings are immutable . . . . . . . . . . . . . . . . . . . . . . . . . . 70

6.6 Looping and counting . . . . . . . . . . . . . . . . . . . . . . . . . . 70

6.7 The in operator . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71

6.8 String comparison . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71

6.9 String methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71

6.10 Parsing strings . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74

6.11 Format operator . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74

## 7 Files p.79

7.1 Persistence . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79

7.2 Opening files . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80

7.3 Text files and lines . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81

7.4 Reading files . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82

7.5 Searching through a file . . . . . . . . . . . . . . . . . . . . . . . . 83

7.6 Letting the user choose the file name . . . . . . . . . . . . . 85

7.7 Using try, except, and open . . . . . . . . . . . . . . . . . . . . . 86

7.8 Writing files . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87

# Schedule for CSC482

|  |  |
| --- | --- |
| **Item** | **Scheduled Due Date** |
| Module 1: Brief Intro to Python & DataCamp | Week 1: 8/26 – 8/30 |
| Module 2: Variables, Expressions and Statements | Weeks 2 & 3: 9/2 – 9/13 |
| Module 3: Conditional Execution | Weeks 4 & 5: 9/16 – 9/27 |
| Module 4: Functions | Weeks 6 & 7: 9/30 – 10/11 |
| Intermediate Presentation | Weeks 8 & 9: 10/14 – 10/25 |
| Module 5: Iteration | Weeks 10 & 11: 10/28 – 11/8 |
| Module 6: Strings | Weeks 11 & 12: 11/11 – 11/22 |
| Module 7: Files | Weeks 13 & 14: 11/25 – 12/6 |
| Final Presentation | Week 15 (Finals): 12/9 – 12/12 |

# References

“Assessment & Instructional Alignment.” *Introduction*, 2007, [www.ucdenver.edu/faculty\_staff/faculty/center-for-faculty-development/Documents/tutorials/Assessment/index.htm](http://www.ucdenver.edu/faculty_staff/faculty/center-for-faculty-development/Documents/tutorials/Assessment/index.htm)

Severance, Charles Russell. Python for Everybody Exploring Data Using Python 3. Create Space Independent Publishing Platform, 2016. <http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf>