**Python Programming[[1]](#footnote-1),[[2]](#footnote-2)**

**The School of Library, Archival and Information Studies (SLAIS)**

**The University of British Columbia**

**Course title**: Python Programming, **Course Code:** LIBR 559C

**Program**: MLIS, Dual

**Year**: Winter Session II 2017

**Time:** 9:00am-11:50am  
**Location**: SLAIS Terrace Lab

**Instructor**: Dr. Muhammad Abdul-Mageed

**Office location**: 494

**Office phone**: 6048-274-530

**Office hours**: Mon. 1:00pm-3:00pm

**E-mail address**: muhammad.mageed@ubc.ca

**iSchool@UBC Student Portal**: <http://connect.ubc.ca>

1. **Course Rationale & Goal:**

**Rationale/Background:** Python is a remarkably powerful dynamic programming language that is used for a very wide variety of application domains, including text processing, data mining and analytics, and web development. Python’s very clear, readable syntax (e.g., use of English keywords where other languages use punctuation) and strong introspection capabilities make it an ideal choice for novice programmers. Indeed, more and more instructors are adopting the language for introductory programming courses.

**Goal:** The goal of this course is to teach programming concepts using the Python programming language. The course is designed to be self-standing and proceeds from basic- to medium- and advanced concepts and assumes no prior programming experience. For an IT professional, programming skills are almost indispensable: General understanding of programming and programming experience both improve students’ overall understanding of information systems and help develop their general problem solving skills. Programming skills are also essential for analysis and mining of information users’ behaviors and communities. Good programming skills significantly boost work productivity and students usually utilize them during their program of study and throughout their professional lives.

Potential audiences for this course are:

* People with little or no programming and IT experience who want to acquire a medium-to-advanced level understanding of programming generally and Python specifically.
* People planning to take (or have taken) courses on text analytics, social media mining, information retrieval, information visualization, human-information interaction, natural language processing, machine learning, web and database design, or similar courses, who want to prepare for, or significantly advance, carrying out work and research in these areas.

1. **Course Objectives**:

Upon completion of this course students will be able to:

* Identify, analyze, assess, and solve a problem or need using Python programming. [1.1]
* Apply Python programing knowledge to practical information access, extraction, retrieval, and data (e.g., text) mining tasks. [1.2]
* Enhance interpersonal and written communication skills through assignments and discussions with classmates and instructor. [2.1]
* Collaborate effectively with peers through course assignments, including demonstrating the ability to work as part of a team, including initiative taking, integrity, dependability and co-operation. [3.1]
* Understand, develop, and apply relevant library and information science tools to address information needs, questions and issues. [4.1]

1. **Course Topics**:

* Overview of (Python) programming;
* Unix:
  + - Directory navigation & control
    - File maintenance commands
    - Display commands
    - Text processing
  + Data types (numbers. Strings, lists, tuples, and dictionaries/hashes);
  + Control flow & functions;
  + File input and output;
  + Exception handling;
  + Iterators and generators;
  + Object-oriented programming;
  + Regular expressions
  + Applications:
    - Web crawling and parsing
    - Text processing
    - Statistics & data visualization

1. **Prerequisites**:
   * + MLIS and Dual MAS/MLIS: Completion of MLIS Core or permission of SLAIS Graduate Advisor
     + Access to a computer: There will be machines in the lab where class is held, but you will need to use your own machine or have access to a machine on a regular basis. You should make your own arrangements for this.
2. **Format of the course**:

* This course will involve lectures, class hands-on activities, individual and group work, and instructor-, peer-, and self-assessment.

**6. Course syllabus:**

***Required book:***

* Chun, W. (2012). *Core python application programming* (3rd ed.). Prentice Hall. (ISBN-13: 007-6092048114; ISBN-10: 0132678209)

***Suggested supplemental books:***

* Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2013). *Data structures and algorithms in Python*. Wiley Publishing. (ISBN-13: 978-1118290279; ISBN-10: 1118290275)
* Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. " O'Reilly Media, Inc.". (ISBN-13: 063-6920516491; ISBN-10: 0596516495)
* Gawron, J. Mark. (2015). *Python for social science.* Available at: <http://www-rohan.sdsu.edu/~gawron/python_for_ss/course_core/book_draft/index.html>. Accessed May 29th, 2016.

***Online Tutorials:***

Pyplot:

* <http://matplotlib.org/api/pyplot_api.html>
* <http://matplotlib.org/users/pyplot_tutorial.html>
* <http://www.scipy-lectures.org/intro/matplotlib/matplotlib.html#beyond-this-tutorial>

Pandas:

* <http://www.scipy-lectures.org/packages/statistics/index.html>

**7. Calendar / Weekly schedule and readings (tentative)**

C = Chun, W. Book

G= Gawron, M. Book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week** | **Date** | **Topic** | **Readings** | |  |
| 1 | Jan 5 | Course overview, Intro. to Unix | Online tutorials, e.g.:   * <https://files.fosswire.com/2007/08/fwunixref.pdf> * <http://www.ee.surrey.ac.uk/Teaching/Unix/> * <http://www.tutorialspoint.com/unix/unix-file-management.htm> | |  |
| 2 | Jan 12 | Intro. to Python; Numbers | C-Ch. 1-5 | |  |
| 3 | Jan 19 | Sequences (Strings, Lists and Tuples) | C-Ch. 6 | | **Assignment 1 out** |
| 4 | Jan 26 | Dictionaries; Conditionals & Loops | C-Ch. 7, 8 | |  |
| 5 | Feb 2 | Files and Input/Output; Errors and Exceptions | C-Ch. 9, 10 | | **Assignment 1 due**  **Assignment 2 out** |
| 6 | Feb 9 | Functions; Modules | C-Ch. 11, 12 | |  |
|  | Feb 16 |  | Winter Break |
| 7 | Feb 23 | Classes | C-Ch. 13 | |  |
| 8 | Mar 1 | Regular expressions | C-Ch. 15 | | **Assignment 2 due** |
| 9 | Mar 8 | Web crawling | G-[Ch. 11](http://www-rohan.sdsu.edu/~gawron/python_for_ss/course_core/book_draft/web/web_intro.html) | | **Project outline due** |
| 10 | Mar 15 | Text processing I | NLTK [Ch. 3](http://www.nltk.org/book/ch03.html) | |  |
| 11 | Mar 22 | Statistics & data visualization I | Pandas tutorials I | |  |
| 12 | Mar 29 | Statistics & data visualization II | Pandas tutorials II & Pyplot | |  |
| 13 | Apr 5 |  |  | | **Presentations** |
|  | Apr 8 |  |  | | **Final paper due** |

**8. Course Assignments:**

|  |  |  |
| --- | --- | --- |
| ***Assignment*** | ***Due date*** | ***Weight*** |
| ASSIGNMENT 1: In-class code | Throughout first 12 weeks | 30% |
| ASSIGNMENT 2: | Feb. 2 | 10% |
| ASSIGNMENT 3: | March 1 | 15% |
| ASSIGNMENT 4: Term Project (GROUP)  Proposal  Presentation  Report/Product | March 8  April 5  April 8 | 5%  10%  30% |

**\* Notes about assignments:**

* All written assignments must be submitted before 11:55 pm of a respective assignment date;
* All coding assignments must be submitted in the form of an iPython notebook. Support will be provided on how to create notebooks;
* Project proposal/outline must be in a pdf format;
* All submitted files should be labeled with your last name(s) followed by an underscore and an assignment code. Assignment codes are two digit numbers, e.g., “assignment01,” “asignment02,” etc.
  + For example: “abdul-mageed\_assignment01.ipynb”

**Assignments in Detail**

***Assignment 1: In-class code***

Throughout the first 12 weeks, you are required to submit the code you write during the class session. This is meant to engage you during the class time and also ensure you are able to run code shared with you during a given session. This assignment will also involve simple extensions of code shared by the instructor. In-class code will be in the form of short code snippets and you will not be penalized for mistakes you make in the code so long as your code runs for 70% of the respective tasks. In-class assignments are only accepted if you attend class. If you miss a class, this automatically means you are not allowed to submit in-class code for the session/sessions you missed. As such, failure to attend class without a necessary excuse (e.g., a sickness or another emergency) will result in a lower class grade.

Deliverables: In-class code snippets

A total of 12 code snippets for the code you write for each of the first 12 weeks of class.

***Assignment 2 & assignment 3: Coding assignments***

These two assignments are coding exercises meant to support learning of the covered content. Each of these will be extensions of problems covered in class. The following criteria are required for each of them:

* You must run your code before submitting, and ensure it is working code. Also, you should provide the output of the code in the iPython notebook itself. For code that writes to a file, you are required to deliver the output files as well.
* For each problem, in addition to your code, you should provide explanatory Python comments as appropriate. Your comments should be short and to-the-point. As a programmer, you should develop a sense as to when to use comments and when not. This requirement is meant to teach you about using comments. Using comments will be discussed in class and feedback on your deliverables will be provided.
* In addition to the functionality of your code, you should care for algorithms and style. This means, over time, you are expected to develop a sense of what the best ways to solve a problem in a computationally efficient way and using the capabilities of the Python programming language. These are aspects that will be discussed in class over the course of the semester.

Deliverables: iPython notebooks with code

* For each of assignment 1 and assignment 2, you are required to submit an iPython notebook including your code and relevant narratives/comments, etc.

***Assignment 4: Term Project***

The purposes of this assignment include:

* Identifying, analyzing, assessing, and solving a problem or need using Python programming;
* Applying acquired Python skills and knowledge to practical tasks;
* Developing oral and written communication skills through discussions with classmates and instructor;
* Demonstrating ability to work as part of a team, including initiative taking, integrity, dependability and co-operation, and effective collaboration.

For this assignment, each student is required to work as part of a group of of 3/4\* on a project involving a significant amount of coding. Example projects include designing a system that generates language (e.g., the language of William Shakespeare), analyzes emotions in a dataset of literary work, visualize concepts from a data source, summarize texts, build a website that delivers some service, build a web interface or an app for a service (e.g., a library service), build an interactive database with use case scenarios in meaningful contexts.

**Deliverables**

Proposal (500-700 words)

* Who are the the group members?
* What are you designing or coding for?
* What motivates your work? Why is it important or useful? For which audience?
* How does your project compare to other projects people have articulated?
* How do you think the project will further your Python programming skills?
* What Python libraries do you expect to use?
* How does the work bread down and what each member of the team be contributing?
* Timeline for completing the project, including goals for each segment.

Presentation

Each team will present their final project to the class. Each student is required to participate in the presentation preparation and articulation. Your presentation should first motivate your work, explaining its significance and audience, and how it furthered your Python programming skills. The team should then explain the various steps the work involved and the final outcome and its functionality. The time devoted to presentations will depend on the number of projects and the class time available. More details about scheduling will be provided and discussed toward the end of the semester.

Final Report/Paper (4000-5000 words)

The final deliverable should include:

* A detailed and clear description of your project, including the necessary sections, as appropriate. For system-oriented projects (e.g., building a website or an app), you will need to provide motivations, describe the audience, details of your implementation, and system functionality. For projects more focused on data analysis, you will need to include research questions, a brief literature review, a description of datasets, implementation details and methods employed, results, and a conclusion involving limitations and future directions;
* All relevant code;
* Pointers to live versions of your system, if any;
* Some of these components will be more relevant to some projects but not others. For example, if your project involves a system for analysis of a dataset, you will need to include standard sections as you would find in an academic journal;
* As appropriate, you should situate your work within the wider context of previous works and approaches, with supporting arguments (~ 10 sources);
* Use case scenarios and screenshots from your system interface, if any;
* Employment of figures, tables, and visualizations as appropriate to enhance argument and facilitate communicating your findings/results;

**9. Course Policies**

**Attendance**: The UBC calendar states: “Regular attendance is expected of students in all their classes (including lectures, laboratories, tutorials, seminars, etc.). Students who neglect their academic work and assignments may be excluded from the final examinations. Students who are unavoidably absent because of illness or disability should report to their instructors on return to classes.”

**Evaluation**: All assignments will be marked using the evaluative criteria given in this syllabus and also those provided on the [SLAIS web site](http://www.slais.ubc.ca/RESOURCES/ischool-policies/letter-grades-and-grading-policy/).

**Written & Spoken English Requirement**: Written and spoken work may receive a lower mark if it is, in the opinion of the instructor, deficient in English.

**Access & Diversity**: Access & Diversity works with the University to create an inclusive living and learning environment in which all students can thrive. The University accommodates students with disabilities who have registered with the Access and Diversity unit: [<http://www.students.ubc.ca/access/drc.cfm>]. You must register with the Disability Resource Centre to be granted special accommodations for any on-going conditions.

**Religious Accommodation**: The University accommodates students whose religious obligations conflict with attendance, submitting assignments, or completing scheduled tests and examinations. Please let your instructor know in advance, preferably in the first week of class, if you will require any accommodation on these grounds. Students who plan to be absent for varsity athletics, family obligations, or other similar commitments, cannot assume they will be accommodated, and should discuss their commitments with the instructor before the course drop date. UBC policy on Religious Holidays: <http://www.universitycounsel.ubc.ca/policies/policy65.pdf>.

**Academic Integrity**

Plagiarism

The Faculty of Arts considers plagiarism to be the most serious academic offence that a student can commit. Regardless of whether or not it was committed intentionally, plagiarism has serious academic consequences and can result in expulsion from the university. Plagiarism involves the improper use of somebody else's words or ideas in one's work.

It is your responsibility to make sure you fully understand what plagiarism is. Many students who think they understand plagiarism do in fact commit what UBC calls "reckless plagiarism." Below is an excerpt on reckless plagiarism from UBC Faculty of Arts' leaflet, "Plagiarism Avoided: Taking Responsibility for Your Work," (<http://www.arts.ubc.ca/arts-students/plagiarism-avoided.html>).

"The bulk of plagiarism falls into this category. Reckless plagiarism is often the result of careless research, poor time management, and a lack of confidence in your own ability to think critically. Examples of reckless plagiarism include:

* Taking phrases, sentences, paragraphs, or statistical findings from a variety of sources and piecing them together into an essay (piecemeal plagiarism);
* Taking the words of another author and failing to note clearly that they are not your own. In other words, you have not put a direct quotation within quotation marks;
* Using statistical findings without acknowledging your source;
* Taking another author's idea, without your own critical analysis, and failing to acknowledge that this idea is not yours;
* Paraphrasing (i.e. rewording or rearranging words so that your work resembles, but does not copy, the original) without acknowledging your source;
* Using footnotes or material quoted in other sources as if they were the results of your own research; and
* Submitting a piece of work with inaccurate text references, sloppy footnotes, or incomplete source (bibliographic) information."

Bear in mind that this is only one example of the different forms of plagiarism. Before preparing for their written assignments, students are strongly encouraged to familiarize themselves with the following source on plagiarism: the Academic Integrity Resource Centre <http://help.library.ubc.ca/researching/academic-integrity>. Additional information is available on the Connect site <http://connect.ubc.ca>.

If after reading these materials you still are unsure about how to properly use sources in your work, please ask me for clarification. Students are held responsible for knowing and following all University regulations regarding academic dishonesty. If a student does not know how to properly cite a source or what constitutes proper use of a source it is the student's personal responsibility to obtain the needed information and to apply it within University guidelines and policies. If evidence of academic dishonesty is found in a course assignment, previously submitted work in this course may be reviewed for possible academic dishonesty and grades modified as appropriate. UBC policy requires that all suspected cases of academic dishonesty must be forwarded to the Dean for possible action.

1. Subject to minor changes without notice. [↑](#footnote-ref-1)
2. Copyright © 2016 Muhammad Abdul-Mageed.  All rights reserved.

   No part of the materials including graphics or logos, available in this material may be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form, in whole or in part, without specific permission (to request permission to use materials, email: [muhammad.mageed@ubc.ca](mailto:muhammad.mageed@ubc.ca)).  [↑](#footnote-ref-2)