Course Syllabus
University of Wisconsin – Milwaukee (UWM)
School of Information Studies (SOIS)

Course Title
Introduction to Application Development

Semester
Spring 2019

Course and Section Number
INFOST 350 – 201
INFOST 350 – 202

Credits
This is a 3-credit course.

Meeting Times and Location
Online

Instructor
Kevin Trainor
Office: NWQB-3472

Teaching Assistant
Brian Hartling

Contacting Instructor or Teaching Assistant
The preferred method for contacting either Brian or me is by entering a request using the [Request Center portal for this course](#). PLEASE, DO NOT send requests to our regular email addresses.

The Request Center portal for this course has been implemented using the [iCourse – JIRA Service Desk](#). If you are new to using the service desk, please visit the [introduction page](#) for instructions and tips.

On an emergency basis, you may contact me using my mobile phone number:
847-650-9706.

Individual Meetings
While I will not be holding scheduled office hours, you can arrange an individual meeting with me by submitting an *Individual Meeting Request* using the Request Center portal (see *Contacting Instructor or Teaching Assistant* above).

Online Lab Sessions
On Sunday mornings from 11:00 AM till 12:00 PM, I will be holding an optional Online Lab Session *using an online meeting platform that will be announced during the first week of the course*. This will be a session in which you can ask questions, discuss ideas, get help with your coding assignments, or get help with your Final Project. I will

work with students on a first come, first served basis. Please use a headset when joining our session.

Catalog Course Description
Introduction to the fundamental concepts of application development. Basic application development concepts will be explained, analyzed, and practiced.

General Course Description
This course introduces the fundamental concepts of application development. This course acquaints students with the core concepts of programming from an Information Studies perspective. Students will learn how to develop basic software applications using the Python programming language that can be applied to further coursework and a career in application development.

Prerequisites
INFOST 110 (C or Better) or instructor consent. Students should have an understanding of basic computer operations, such as typing, accessing the internet, and technology troubleshooting skills.

Course Outcomes
After completing this course, you should be able to:

1. Explain fundamental programming concepts.
2. Create working application logic to conduct basic tasks using a programming language.
3. Verbally discuss and explain code and application logic to other students.

BSIST Learning Outcomes
1. Design, implement, evaluate, and administer information systems to meet a given set of requirements – including user and organizational needs -- in diverse environments.
3. Communicate effectively in a variety of professional contexts.

Instructional Methods
- Reading
- Recorded lectures and tutorials
- Text-based online discussion via D2L forums
- Voice-based online discussion via optional Online Lab Sessions on GoToMeeting
- Designing, coding, and testing Python programs

Time Expectations
This course requires a weekly time commitment. General university guidelines indicate that a 3-credit course requires a minimum 144-hour time commitment over the semester. This time commitment represents a minimum of 9-10 hours of work per week per course. Approximately 3 of these hours are recorded lectures and tutorials. Students are expected to do an additional 6-7 hours per week of study and work on assignments to achieve the learning goals of this course.

Required Texts

Zelle, J. (2016). Python Programming: An Introduction to Computer Science, 3rd Ed
ISBN: 9781590282755

Severance, C. (n.d.). Python for Everybody (https://www.py4e.com/book).
This text is available as an electronic book in several formats. Although it is also available as a printed book, I DO NOT RECOMMEND PURCHASING a printed copy. I invite you to take advantage of this author's generosity by using one of these electronic formats.

Technology Requirements
I highly recommend that you use your own computer for this course. If you do not have a computer available to you, you will need to use the SOIS lab computers. If you are trying to use a SOIS lab computer for this course and having difficulty with the setup, please contact the SOIS Tech Help Desk.

You will be completing coding assignments and conducting your final project using your own computer. I recommend that you use a computer that runs Windows 10 or a recent release of MacOS. While the software that we will be using for this course does run on Linux computers, there will be substantially less technical support available for Linux. If you want to use a Linux computer for your coursework, please contact me first.

You will be installing the following software on your computers for this class:

- The Anaconda open data science platform for Python
- The PyCharm integrated development environment (IDE) for Python

All of the software and services that you will be using are free for your use during this class. I will publish brief instructions and detailed tutorial videos to help you download, install, and begin using the required software on computers running Windows 10 and recent releases of MacOS. Links to these resources will be provided in our Weekly Schedule.

Course Schedule
The schedule for this course will be available via our Weekly Schedule at:

https://courseinfo.ligent.net/2019fa/_uwm/infost350_201_202/index.html

The course schedule is always subject to reasonable change to account for changes in circumstance and to correct errors. When I make changes to the schedule, I will announce them via Canvas Announcements and email.

Course Elements

1. Readings

Required readings will be assigned from the text books listed above and from other resources that will be identified in the Weekly Schedule. Generally, readings are chosen to accompany any lecture video or tutorial video provided for the week. So, you should expect to complete the readings before playing the videos.

This course has two required text books (Zelle and Severance). Readings in the Zelle text are marked in the Weekly Schedule as required. Most readings in the Severance text are marked in the Weekly Schedule as optional. While Zelle and Severance cover the same Python programming material, they take different approaches. I have found that reading a second approach to the same subject is particularly helpful for first time programmers and for anyone having difficulty with a particular topic. I expect you to use your best judgement in deciding whether the optional readings are adding appropriate value for you.

2. Lecture Videos

I have provided links to pre-recorded video lectures for each unit that we cover. While most of the material covered in the recorded lectures is from the textbook, I occasionally cover supplemental material in the lectures as well. I always include commentary that I believe adds value to the text.

Before playing my lecture videos, make sure that you have previously played *Tips on Playing My YouTube Videos* to assure that you get the most from your viewing experience.

3. Tutorial Videos

I have created tutorial videos for the following use cases:

- There are a number of tutorials that explain how to do activities necessary for the course. These include installing software, reading grading rubrics, submitting assignments, and related activities.

- Many tutorials are coordinated with the exercises in your coding assignments. They represent a demonstration of the skill you will be expected to use when doing the exercise.

Because tutorial videos do not duplicate material covered by other resources, I recommend that you always play these recordings. Before playing my tutorial videos, make sure that you have previously played *Tips on Playing My YouTube Videos* to assure that you get the most from your viewing experience.

4. *Coding Assignments*

There will be weekly Coding Assignments. As mentioned above, coding assignments will often be paired with tutorial videos. These should allow you to complete your Coding Assignment using the same general approach that has been demonstrated in the video.

A Canvas assignment submission activity will be provided for submitting each assignment. Instructions for each Coding Assignment and a grading rubric will be published in the Weekly Schedule.

Solutions to Coding Assignments will be posted to our Canvas site soon after the due date and time of the assignment. I will review those solutions at the beginning of the next Online Lab Session.

A major goal for this course is to build your proficiency in self-evaluation of your work. To build this skill, I will expect you to be able to estimate your grade on each Coding Assignment. The solutions to coding assignments posted to our Canvas site and my review of those solutions during the next Online Lab will serve as your primary feedback for the Coding Assignment.

As secondary feedback, your Coding Assignment submissions will be graded and commented upon. This feedback will be published to the D2L assignment submission activity within 2 weeks.

The grading rubric for Coding Assignments has been designed to promote two important behaviors:

- Submitting your work in a properly named and formatted file. This helps substantially with grading workflow.

- Submitting your work by the week-ending deadline. This assures that you will get the benefit of having tried to solve the problem on your own before seeing the solutions of others.

While separate grading rubric and assignment submission instructions documents will be published, the following is a summary of the coding assignment grading rubric features:

- 10 points will be awarded for submitting a single, properly named and properly formatted file to the proper Canvas assignment submission activity.

- A minimum of 75 points will be awarded for submissions that are submitted on time, and that demonstrate a good faith effort on all parts of the assignment. Late submissions will be awarded 74 points or fewer in this category.

The implication of this grading scheme is that you can expect a score of 85 or higher on all Coding Assignments that meet both of these criteria.

5. *Participation*
   a. You are expected to contribute a significant (200 – 300 word) post to the *Introduce Yourself Forum* during Week 1 of the semester.

b. You are expected to contribute a significant (200 – 300 word) post to the *Next Steps Forum* during Week 10 of the semester.

c. You are also expected to read all of the posts of other students made in all of the discussion forums and respond with short posts when appropriate.

d. For each participation exercise (see *a* and b), a maximum of 50 points may be earned. Twenty-five points will be earned for your own primary post. Another 25 points will be earned for a minimum of 2 significant responses to the primary posts of other students.

6. *Final Project*

The Final Project will be a significant multi-part assignment to be completed in the final weeks of the semester. This assignment will require you to design, code, and test an application solution consisting of multiple programs and using techniques covered throughout the course. The instructions and grading rubric for the Final Project will be published closer to the end of the course.

Grading

Basis for Determining Grade
The following weights will be used when computing final number grades:

Grading Component	Grade Weight
Coding Assignments	60%
Final Project	35%
Participation	5%
Total	100%

Letter grades will be determined as follows:

- A   93 - 100%;
- A-   90 - 92%;
- B+   87 - 89%;
- B   83 - 86%;
- B-   80 - 82%;
- C+   77 - 79%;
- C   73 - 76%;
- C-   70 - 72%;
- D+   67 - 69%;
- D   63 - 66%;
- D-   60 - 62%;
- F   0 - 59%;

UWM AND SOIS ACADEMIC POLICIES

The following links contain university policies affecting all SOIS students. Many of the links below may be accessed through a PDF-document maintained by the Secretary of the University: http://www.uwm.edu/Dept/SecU/SyllabusLinks.pdf. Undergraduates may also find the *Panther Planner and Undergraduate Student Handbook* useful (http://uwm.edu/studenthandbook/student-handbook/).

Students With Disabilities
If you will need accommodations in order to meet any of the requirements of a course, please contact the instructor as soon as possible. Students with disabilities are responsible to communicate directly with the instructor to ensure special accommodation in a timely manner. There is comprehensive coverage of issues related to disabilities at the Student Accessibility Center (http://www4.uwm.edu/sac/), important components of which are expressed here: http://www.uwm.edu/Dept/DSAD/SAC/SACltr.pdf.

Religious Observances
Students' sincerely held religious beliefs must be reasonably accommodated with respect to all examinations and other academic requirements, according to the following policy: http://www4.uwm.edu/secu/docs/other/S1.5.htm. Please notify your instructor within the first three weeks of the Fall or Spring Term (first week of shorter-term or Summer courses) of any specific days or dates on which you request relief from an examination or academic requirement for religious observances.

Students Called to Active Military Duty
UWM has several policies that accommodate students who must temporarily lay aside their educational pursuits when called to active duty in the military (see http://www4.uwm.edu/academics/military.cfm), including provisions for refunds, readmission, grading, and other situations.

Incompletes
A notation of "incomplete" may be given in lieu of a final grade to a student who has carried a subject successfully until the end of a semester but who, because of illness or other unusual and substantial cause beyond the student's control, has been unable to take or complete the final examination or some limited amount of other term work. An incomplete is not given unless the student proves to the instructor that s/he was prevented from completing course requirements for just cause as indicated above (http://www4.uwm.edu/secu/docs/other/S31.pdf).

Discriminatory Conduct (such as sexual harassment)
UWM and SOIS are committed to building and maintaining a campus environment that recognizes the inherent worth and dignity of every person, fosters tolerance, sensitivity, understanding, and mutual respect, and encourages the members of its community to strive to reach their full potential. The UWM policy statement (http://www4.uwm.edu/secu/docs/other/S47.pdf) summarizes and defines situations that constitute discriminatory conduct. If you have questions, please contact an appropriate SOIS administrator.

Academic Misconduct
Cheating on exams and plagiarism are violations of the academic honor code and carry severe sanctions, ranging from a failing grade for a course or assignment to expulsion from the University. See the following document (http://uwm.edu/academicaffairs/facultystaff/policies/academic-misconduct/) or contact the SOIS Investigating Officer (currently the Associate Dean) for more information.

Complaints
Students may direct complaints to the SOIS Dean or Associate Dean. If the complaint allegedly violates a specific university policy, it may be directed to the appropriate university office responsible for enforcing the policy.

Grade Appeal Procedures
A student may appeal a grade on the grounds that it is based on a capricious or arbitrary decision of the course instructor. Such an appeal shall follow SOIS appeal procedures for undergraduates as seen here: (http://www4.uwm.edu/sois/programs/graduate/mlis/policies/appeals.cfm) In the case of a graduate student, the Graduate School, (http://www4.uwm.edu/sois/programs/undergraduate/ug_appeals.cfm).

Last Revised
2019-09-02