# CYBR 2980 Special Topics in CYBR – Python Programming for Interdisciplinary Informatics

Spring 2020

## Syllabus

|  |  |
| --- | --- |
| Time | Monday & Wednesdays 10:30 – 11:45 AM |
| Location | PKI 361 |
| Instructor | Derek Babb  Office: 283C  Email: dvbabb@unomaha.edu |
| Course Overview | This course will support Cybersecurity and Si2 curricular efforts to equip students with practical scripting, visualization, and analysis skills relevant to their program curricula. The course will be designed around a number of extensible, updatable, modules. Initially, the course will target students who only need one programming course in their major, but is designed to accommodate cybersecurity, bioinformatics, and ITIN modules to be developed by the cognizant program committees at a later date. |
| Prerequisites | CSCI 1200 or CIST 1300 |
| Course Text | None |
| Academic Integrity | Helping other students on assignments is generally encouraged, but students must author and complete their own work. No cooperation or sharing is permitted on examinations.  If you are not sure about the difference between permitted cooperation and prohibited misconduct for a situation, you must acquire clarification from the Instructor.  Every submittal which is determined by the Instructor to be a copy of another submittal will be scored as zero. It is your responsibility to protect your work from being copied. It is your responsibility to clean your workstation so that subsequent users of that workstation cannot copy your work.  Allowing your work to be copied is considered a failure on the assignment, but may not be academic misconduct. Copying another student's work and/or misrepresenting another's work as your own is academic misconduct. The procedures and sanctions for incidents of academic dishonesty, academic misconduct, and other violations of academic integrity will be governed by UNO policies as documented in the publication "Undergraduate Academic Integrity" available in the UNO IS&T Dean's Office. |
| Attendance | Classroom attendance and participation is expected and required. If a student is unable to attend a class, it is their responsibility to learn the material covered in the missed class. |
| Assignments | Multiple programming assignments will be used throughout the course to learn and demonstrate an understanding of the course material. Class time will be given for lab exercises. |
| Retained Copies | The Bachelor of Science Information Systems (BIS) and Bachelor of Science Computer Science (BCS) Programs are accredited by ABET, the accreditation board for engineering and technology. This organization requires that we keep samples of student work. Unless you specify otherwise, I may retain the original copy of your exams and assignments for accreditation purposes and return a copy to you. |
| Grading | Course grading is determined by student performance with respect to participation, programming assignments, and exams. Late assignments will be accepted but penalized up until when the solution is made available to students. Assignments cannot be accepted after the solution is made available to the class through lecture, handout, or posting to Canvas.  The grading scale is A+ through F.  Grading components, subject to change during the course with prior notice to students:     * Labs 20% * Lecture Quizzes 5% * Programs 40% * Midterm Exam 15% * Final Exam 20%   Upon written request from the student, instructor will review grades assigned to any component and/or the final grade. |
| Key dates | Consult the academic calendar at www.unomaha.edu for deadlines such as: course drops; changing from graded to CR/NC or audit; and applying for degrees. |
| Disabilities | Accommodations are provided for students who are registered with Disability Services and make their requests sufficiently in advance. For more information, contact Disability Services (EAB 117, Phone: 554-2872, TTY: 554-3799) or visit the website: www.unomaha.edu/disability. |

## Schedule

The following schedule is subject to change.

|  |  |
| --- | --- |
| Week | Topic |
| 1 | Introduction - Topics in computer science, algorithms; modern  computer systems: hardware architecture, data representation in computers,  software and operating system.  Installing Python, basic syntax, interactive shell, editing, saving, and running  a script. |
| 2 | The concept of data types; variables, assignments; numerical types; arithmetic operators and expressions; comments in the  program; understanding error messages; |
| 3 | Conditions, boolean logic, logical operators; ranges;  Control statements: if-else, loops (for, while); short-circuit evaluation |
| 4 | Turtle Graphics, algorithms |
| 5 | String manipulations: subscript operator, indexing, slicing a string; strings  and number system: converting strings to numbers and vice versa. |
| 6 | Lists, tuples, and dictionaries; basic list operators, replacing, inserting,  removing an element; searching and sorting lists; dictionary literals, adding  and removing keys, accessing and replacing values;  traversing dictionaries. |
| 7 | Design with functions: hiding redundancy, complexity; arguments and return  values; formal vs actual arguments, named arguments.  Program structure and design. |
| 8 | Midterm Exam |
| 9 | Strings and text files; manipulating files and directories, os and sys modules;  text files: reading/writing text and numbers from/to a file; creating and  reading a formatted file (csv or tab-separated). |
| 10 | Image processing and manipulation; image formats, filter functions, steganography. |
| 11 | Spring Break |
| 12 | Cryptography tools in Python |
| 13 | Data Visualization in Python using Matplotlib, Pandas |
| 14 | Python HTTP interfaces |
| 15 | Python Tools: SCAPY, Simpy |
| 16 | Searching, Sorting, and Complexity Analysis  Linear Search, Binary Search, Bubble Sort, Selection Sort |
| 17 | Cleanup and Final Review |
| 18 | FINAL |