UNI CS 1510 (Spring 2020)
Introduction to Computing, Section 01

Course Syllabus

Course Information

Course Name and Number
CS 1510 Introduction to Computing

Meeting Times
Lecture (Section 01): MWF 10:00am–10:50am 322 ITTC
Lab (Section 01): Th 8:00am–9:50am 112 Wright

Contact Information
(Lead Professor) Sarah Diesburg – sarah.diesburg@uni.edu
Office: 311 ITTC Building
Office hours: TBD
Class website: Available through UNI eLearning: https://elearning.uni.edu

(Lab Instructor) Mark Jacobson – mark.jacobson@uni.edu
Office: WRT 338a
No office hours

Credit Hours
4 Credit Hours (3 for lecture, 1 for lab)
This course meets the Course Credit Hour Expectation outlined in the Course Catalog. Students should expect to work approximately 2 hours per week outside of class for every course credit hour.

Course Materials (Required)
Lecture notes (posted on the class Web site)
Follow these directions to obtain access to your mandatory online textbook:
  o Sign in or create an account at http://learn.zybooks.com
  o Enter zyBook code: UNICS1510DiesburgSpring2020
  o Subscribe

Course Description
Introduction to software development through algorithmic problem solving and procedural abstraction. Programming in the small. Fundamental control structures, data modeling, and file processing. Significant emphasis on program design and style. (Fall and Spring)

Course Learning Outcomes
1. Students can analyze a programming problem and create a programming solution using the basics of programming:
   a. actions (input, processing, output)
b. control structures (selection, repetition)
c. basic data structures (strings, arrays/lists, hash tables/dictionaries)
d. modularization (student-written procedures/functions)
e. data persistence (file input and output)

2. Students will increase their problem solving skills by creating programming solutions to solve real-world problems.

3. Students will increase their communication skills by writing readable programs suitable for collaboration with others.

Performance Evaluation

Grade Determination
The final grade you earn in this course will be based on the points accumulated over five activities as described below.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Quantity</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab Work</td>
<td>13 @ 10 pts each</td>
<td>130</td>
</tr>
<tr>
<td>Individual Homework</td>
<td>11 @ 25 pts each</td>
<td>275</td>
</tr>
<tr>
<td>Concept Exams</td>
<td>125 and 150 points</td>
<td>275</td>
</tr>
<tr>
<td>Programming Exams</td>
<td>2 @ 125 pts each</td>
<td>250</td>
</tr>
<tr>
<td>zyBook Participation Activities</td>
<td>70 pts</td>
<td>70</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>1000</strong></td>
</tr>
</tbody>
</table>

To continue on to the next class in the computer science major, you must earn at least a C.

Grading Scale
100 – 92     A     77.9 – 72     C
91.9 – 90     A-    71.9 – 70     C-
89.9 – 88     B+    69.9 – 68     D+
87.9 – 82     B     67.9 – 62     D
81.9 – 80     B-    61.9 – 60     D-
79.9 – 78     C+    59.9 – 0      F

Class Attendance and Participation
Class attendance is required. If you miss a class, it is your responsibility to find out what was covered.

ZyBook Participation and Challenge Activities
Your textbook is an interactive online textbook. You will be graded for completing participation activities before class on the day they are assigned. Again, participation activities are due before class. Challenge activities are not graded. They are good practice, and we will often go over them in class as sample problems.

In-lab work
Lab is designed to be a time to allow you to learn new skills, apply and practice existing skills, and prepare yourself for the upcoming lectures and programming assignment. Points for these activities will be assigned based on level of difficulty for each activity and will be awarded for successful completion and/or effort.

Attendance to lab sessions is required - you will receive credit only for labs you attend. In general, students who do not show up at all will not receive credit for making up the lab (although you should still complete the activities so you do not fall behind). Lab makeups will be allowed in the cases of university-approved absences.
Programming Assignments
Programming assignments are designed to take what you have learned in lab and during lecture, and apply these skills to a program on a scale larger than that explored in-lab. It is expected that you will complete all assignments as an individual unless otherwise instructed (see section on scholastic conduct). If you have questions concerning an assignment, feel free to consult an instructor, come to office hours, or consult a class TA.

All assignments are due at their assigned date and time. Assignments can be submitted a maximum of two days late, and each day late incurs a 10% penalty. Assignments will be submitted on the Autolab grading system. Using the grading system, you can get immediate feedback and resubmit non-working code.

Description of Assignments
- Create a series of basic programs that (1) ask for input, (2) make calculation(s), and (3) display output.
- Create a series of programs that use selection statements to make choices based on input.
- Create a series of programs that use looping (repetition) to solve problems.
- Create program(s) that use nested looping (repetition) to solve problem(s).
- Create program(s) that demonstrate the use of advanced string functionality to solve problem(s).
- Create program(s) that illustrate the use of functions/procedures to solve problem(s).
- Create program(s) that read data from an external file, use data manipulation, and write output to a new file.
- Create program(s) that use lists/arrays as an internal data structure to solve problem(s).
- Create program(s) that use dictionaries/hash tables as an internal data structure to solve problem(s).
- Create a program design document for a program that involves many of the above components to solve a problem.
- Create programs that use many or all of the above elements to solve problem(s).

Homeworks will by default be assigned after the lab period for which the new concept will be introduced. Please consult the main website for exact homework due dates and late policies.

Exams
There are a total of four exams this semester.
- Two will be concept exams offered during the lecture part of the course.
- Two will be programming exams offered during the lab portion of the course.

By default these exams are closed-book/closed-notes exams. The dates of these exams are listed on the class schedule. You are expected to be present for these exams unless you have made prior arrangements. Make-up exams will be offered under very limited circumstances. If you are aware of conflicts prior to the exam, please bring these to my attention as early as possible.

Incompletes
Incompletes are awarded only in very rare instances when an unforeseeable event causes a student who has completed all the coursework to date to be unable to complete a small portion of the work in the last week or two of the semester (typically the final project or exam). Incompletes will not be awarded for foreseeable events including a heavy course load or a poorer-than-expected performance. Verifiable documentation must be provided for the incomplete to be granted.
Tentative Schedule (Subject to Change)

<table>
<thead>
<tr>
<th>Date</th>
<th>Topics/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/13 – 1/22</td>
<td>Computer Basics, Numerical Representation</td>
</tr>
<tr>
<td>1/22 – 1/27</td>
<td>Intro to Basic Python</td>
</tr>
<tr>
<td>1/29 – 2/3</td>
<td>Booleans, Conditionals, Selection Statements</td>
</tr>
<tr>
<td>2/5 – 2/17</td>
<td>Repetition</td>
</tr>
<tr>
<td>2/19 – 3/2</td>
<td>Basic Strings</td>
</tr>
<tr>
<td>3/4</td>
<td>Concepts Exam #1 in SAB 102</td>
</tr>
<tr>
<td>3/5</td>
<td>Programming Exam #1 in WRT 112</td>
</tr>
<tr>
<td>3/6 – 3/9</td>
<td>File Basics</td>
</tr>
<tr>
<td>3/11 – 3/23</td>
<td>Functions</td>
</tr>
<tr>
<td>3/25 – 3/30</td>
<td>Basic Lists</td>
</tr>
<tr>
<td>4/1 – 4/9</td>
<td>Functional Decomposition / Design</td>
</tr>
<tr>
<td>4/10 – 4/17</td>
<td>Dictionaries</td>
</tr>
<tr>
<td>4/20 – 4/27</td>
<td>Sets, Searching &amp; Sorting Algorithms</td>
</tr>
<tr>
<td>4/30</td>
<td>Programming Exam #2</td>
</tr>
<tr>
<td>5/4</td>
<td>Final: Concepts Exam #2 in SAB 102 (10:00am-11:50am)</td>
</tr>
</tbody>
</table>

For your specific final date and time, please consult the registrar’s final calendar at: https://registrar.uni.edu/calendars/final-examination-schedule

Computing Environment

The following labs have pre-configured software for this class:

- **Wright 112** – This is where you will meet for your lab sessions. This is a public lab part of the week but it also used by other classes at other times of the day/week and may not always be available. It generally closes at 5pm on weekdays.
- **Wright 339** – This lab is open the latest on weekdays (until 9:00pm or so).
- **ITTC 335** – This is a small general purpose lounge available to students in the CS department. This is a good place to get a quick printout or check your email between classes. It generally closes at 5pm on weekdays (or when the last faculty member leaves).

**Working on your own laptop/computer:** You are actually encouraged to work on your own laptop or computer. Having your own computer will greatly aid you in the computer science major, and the computer/laptop itself does not have to be very expensive. The class software is free and will work with Windows, OSX, and Linux. Python and IDLE are easily downloaded from [www.python.org](http://www.python.org). You should download the latest edition of version 3 (NOT version 2).

Whether you work in the labs or from home, you will need to have Internet access to submit your assignments.
Scholastic Conduct

Since cheating definitions and academic ethics policies are often written for other types of classes, you might tend to wonder how those translate to a computer science course. You may be surprised to hear there are many ways to write a program to solve a specific problem. This is very similar to how there are many different ways to write an essay addressing a particular topic. After a certain point in the course, I will be using plagiarism-detection software to detect similarities that are very unlikely to occur if students were working alone.

Additionally, you need to cite your source if you seek and use help found on the Internet (much like citing a source in an essay course). To do this, you need to put the URL and a brief description of the help you found in a comment directly above the affected block of code. I will show you how this is done further along in the class. However, if you do use code from the Internet, I reserve the right to ask you how it works line-by-line. If you cannot explain it to me, I will not give you credit for that part of the assignment. In other words, *if you use help or code found on the Internet, you must cite it and fully understand it*. You never should use code you don’t understand.

In this class, homework assignments must be done on your own as your own individual work. However, this does not mean that you cannot ask for help. Here are some general guidelines for keeping out of trouble.

If you are seeking help from a classmate:
- **DO NOT** ask to see their code or look at their code.
- **DO** explain your thought process and where you are stuck in words.
- **DO** draw diagrams on the board.

If you are helping another classmate:
- **DO NOT** show them your code.
- **DO NOT** directly modify their code.
- **DO** try to help them in words, similar examples from lectures and labs, and diagrams.

Finally:
- **DO NOT** copy code from the Internet without citations. If you do copy code from the Internet without citations, it is considered cheating and at least a Level One violation of the UNI Academic Ethics Policy. This includes “tutoring” websites like Chegg. (If your citation(s) cover more than just a few lines of code, be prepared to meet with me to explain each line verbally to me.)

If I suspect a case of plagiarism or cheating, I will notify the student via email and allow the student to come in and explain what happened. If I determine that plagiarism or cheating has taken place, the following possible sanctions will occur (in accordance with UNI Academics Ethics Policies found at [http://www.uni.edu/policies/301](http://www.uni.edu/policies/301)). The following list does not list all possible academic ethics violations, and it is your responsibility to be familiar with the full list (again, [http://www.uni.edu/policies/301](http://www.uni.edu/policies/301)).

Remember: Discussing assignments is good. Copying code or answers is not.

Student Accessibility

The University of Northern Iowa (UNI) complies with the Americans with Disabilities Act Amendments Act of 2008 (ADAAA), Section 504 of the Rehabilitation Act of 1973, the Fair Housing Act, and other applicable federal and state laws and regulations that prohibit discrimination on the basis of disability. To request accommodations please contact Student Accessibility Services (SAS), located at ITTC 007, for more information either at (319) 273-2677 or Email accessibility services@uni.edu. Visit Student Accessibility Services ([https://sas.uni.edu/](https://sas.uni.edu/)) for additional information.
Non-discrimination Statement
The University of Northern Iowa does not discriminate in employment or education. Visit 13.03 Equal Opportunity & Non-Discrimination Statement (https://policies.uni.edu/1303) for additional information.

The Learning Center @ Rod Library syllabus statement
All students are encouraged to use The Learning Center @ Rod Library for assistance with writing, math, science, and college reading and learning strategies. Beginning week two, The Learning Center (TLC) operates on a walk-in basis and is open 10:00 am to 10:00 pm Monday through Thursday. For more information, go to https://tlc.uni.edu/tutoring, email TheLearningCenter@uni.edu, call 319-273-6023, or visit the TLC desk located on the main floor of Rod Library. If you are unable to come in during normal tutoring hours, online tutoring is available through Smarthinking. You will need your CATID and passphrase to gain access. To access the Smarthinking platform go to https://tlc.uni.edu/schedule.