CSC 110 Course Syllabus

CSC 110 – Computer Science Principles: The Beauty and Joy of Computing

Section TBD

Fall 2019

3 Credit Hours

Course Description

This course explores the principles of computer science while emphasizing the relevance of computing to students and to society. Students will learn about beautiful computing applications that have changed the world and how computing empowers discovery and innovation. Students will learn the joy of programming a computer using a friendly, graphical language, capable of creating apps, simulations, and games. Students who complete the course will be able to solve meaningful problems with computers, apply design processes to take an idea from concept to implementation, develop a computer program, and analyze computing artifacts from both design and computing perspectives. Students will complete a substantial team programming project related to their interests. 20% of seats will be restricted to Computer Science or CS-Intended students. Enrolling students must not have received credit for or a grade in CSC 116 or CSC 200.

Learning Outcomes

The goal of this course is to introduce students to 7 central ideas of computer science and computational thinking and how these ideas shape the world. Students who complete this course will be able to:

Creativity:

- [Interdisciplinary Perspectives] Develop programs that implement specific design goals, including user interfaces, games, and generative art
- [Interdisciplinary Perspectives] Integrate a creative design process with a software design process to collaboratively develop a program

Abstraction:

- Use multiple levels of abstraction to manage the complexity of a program

Data & Information:

- Explain how data are represented, stored, and transmitted on computers and the internet
- Extract information from data, and use this data to find patterns, and test hypotheses to gain insight and knowledge

Algorithms:

- Explain the difference between algorithms that run in a reasonable time and those that do not
- Develop an algorithm and express that algorithm as a program
- [Interdisciplinary Perspectives] Evaluate computer programs from both an analytical perspective (e.g. efficiency, use of abstractions and algorithms) and a design perspective (e.g. usability, functionality)

Programming:

- Use functions, variables, lists, loops, and conditionals appropriately in a program

Internet:

- Explain characteristics of the Internet and the systems built on it
- List two examples of cybersecurity concerns and identify potential options to address them

Global Impacts:

- [Interdisciplinary Perspectives] Explain how design choices in a computing innovation affect society, economy, or culture in both beneficial and harmful ways
- Explain the connections between computing and real-world contexts, including economic, social, and cultural contexts

Course Structure

Classes will consist of a combined lecture/lab, in which students typically work in pairs to complete guided programming activities. Instructors will use these activities as a platform to explore larger ideas, such as abstraction, data, creativity, and the social implications of computing. Students should expect weekly programming assignments, to be completed outside of class. The course will feature two substantial team programming projects (midterm and final projects), in which students design and create a program to solve a problem or to express creativity.

Instructors

Dr. Thomas Price (twprice) - *Instructor*
Email: twprice@ncsu.edu
Web Page: go.ncsu.edu/twprice
Phone: 919-515-1286
Office Location: Venture II, Room 53015
Office Hours: TBD

Dr. Tiffany Barnes (tmbarnes) - *Instructor*
Email: tmbarnes@ncsu.edu
Web Page: http://people.engr.ncsu.edu/tmbarnes
Phone: 919-515-5764
Office Location: Venture II, Room 53019
Office Hours: TBD

Course Meetings

Lecture

Days: TH
Time: 1:30pm - 2:45pm
Campus: Main
Location: TBD
This meeting is required.

Meeting Notes

TBD

Course Materials

Textbooks

Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion - *Hal Abelson, Ken Ledeen and Harry Lewis*

Edition: 1st Edition
ISBN: 978-0137135592
Web Link: http://www.bitsbook.com/
Cost: Free
This textbook is required.

Expenses

None.

Materials

None.

Requisites and Restrictions

Prerequisites

None.

Co-requisites

None.

Restrictions

Students may not enroll in this course if they have credit for or a grade in CSC 116 or CSC 200.

General Education Program (GEP) Information

GEP Category

Interdisciplinary Perspectives

GEP Category Outcomes

Course objectives related to Interdisciplinary Perspectives:

Objective 1: Distinguish between the distinct approaches of two or more disciplines

- Explain how design choices in a computing innovation affect society, economy, or culture in both beneficial and harmful ways

Objective 2: Identify and apply authentic connections between two or more disciplines

- Evaluate computer programs from both an analytical perspective (e.g. efficiency, use of abstractions and algorithms) and a design perspective (e.g. usability, functionality)
- Develop programs that implement specific design goals, including user interfaces, games, and generative art

Objective 3: Explore and synthesize the approaches or views of the two or more disciplines

- Integrate a creative design process with a software design process to collaboratively develop a program

How This Course Will Fulfill GEP Category Outcomes

Objective 1:

The Explore Task writing assignment asks students to explain how a computing innovation has impacted society, economy, or culture, requiring students to identify beneficial and harmful impacts. Students will be asked to consider the impact of these innovations from both the perspective of a computer scientist, explaining how the innovation uses data and how this use relates to security, and the perspective of a designer, evaluating how the innovation impacts users' experiences and behaviors. In the essay, students will be asked to consider the relationship between these two evaluation criteria, when they align and when they conflict. This is a 2-page essay where students must use references, and make well-supported arguments and claims about the connections between computing and its impacts.

Objective 2:
During Create Task 1 (midterm project), students will collaborate to create a new program to accomplish a meaningful task (e.g. an app, simulation, or game). Part of the Create Task 1 project deliverables will include a design critique and a code review of another team's project. In the design critique, students will use each other's software, identify positive and negative elements, and constructively articulate why these elements are effective or ineffective. In the code review (a common software engineering practice), students will look at the program code that makes up their peers' software and make constructive suggestions for how the efficiency and readability could be improved. Afterwards, students will be asked to reflect on how these two perspectives connect and complement each other, with questions such as, "What did you learn about your program from the design critique that you would not have learned from code review, and vice versa?" Students will record both design and software feedback, and part of students' peer evaluation grades will come from the quality of their own peer reviews. Students will then apply this feedback to their project before submitting. Instructors will cover the basic principles of effective design critique and code review during the "Creative Design Process" lecture.

During Labs, students will regularly develop programs that implement specific design goals, including user interfaces, games, and generative art. For example, the Daisy Design lab will require students to create code to draw a daisy design on the computer screen. Creating and implementing a design specification represents one of the most authentic connections between the disciplines of design and computer science. Computer scientists working in industry regularly collaborate with designers to bring their designs to life through a computer program. Each programming lab requires students to carry out this process. Students must read a description of the required elements and the desired output of a program to form a design specification for the lab. This process involves reading and interpreting the requirements in the instructions, imagining the sequence of images that the program should create on the screen and what causes transitions between them (a storyboard), and deciding when and how the user should interact with the program. For more open-ended labs, students will be asked to explicitly record and submit this design specification. Students must then implement their design specification using the tools of computing. They must decide which program elements they will use to make the program, how those elements should be sequenced to make an algorithm, and how they will store and transform data to accomplish those goals. Creating a program that meets the lab’s specifications means that the students have successfully applied the connections between design and computer science, and we will use these labs to assess that ability.

Objective 3:

Students will complete a final team programming project, Create Task 2, in which they create an interactive program to accomplish a meaningful task (e.g. an app, simulation or game) over the course of 3-4 weeks. As part of this project, students will undertake a creative design process, in which they define design goals, identify and critique existing software that addresses these goals, storyboard the user experience, and produce iterative prototypes. Students will record this process in a "design notebook." They will simultaneously be engaging in the software design process, planning out their projects using computing tools including pseudocode, flowcharts, and/or appropriate abstractions. These two design processes will overlap, as students iterate between designing and implementing their prototypes and final program. Students will have time during class to work on these projects in groups, where instructors will give feedback on both design and software goals. These projects will be evaluated based on both the quality of students' design notebooks and that of their program code. Note that while projects will only require students to use introductory programming concepts, students will be required to design and implement programs with sufficient complexity to experience the costs and benefits of collaboration and design processes.

Which disciplines will be synthesized, connected, and/or considered in this course?

The course will focus on a synthesis of design and computer science. However, course readings and lectures will also consider the impact that well-designed computing innovations have had on other disciplines (e.g. advancing by scientific breakthroughs and enabling data science) and on how people think, communicate, live, and work.

How will the instructor present the material so that these disciplines are addressed in a way that allows the students "to integrate the multiple points of view into a cohesive understanding"?

This "Beauty and Joy of Computing" course takes its name because it emphasizes how code can be a creative and delightful process. The interdisciplinary nature of this course will invite students to consider two lenses through which to view the beauty of computing: a design lens that appreciates the beauty of usable and functional software products, and a software lens that appreciates the beauty of elegantly written code and abstractions. The instructor will present and integrate these perspectives through lectures, readings, weekly "Computing in the News" forum posts, hands-on lab assignments, and group projects.

GEP Co-requisites

This course does not fulfill a General Education Program co-requisite.

Transportation

This course will not require students to provide their own transportation. Non-scheduled class time for field trips or out-of-class activities is NOT required for this class.

Safety & Risk Assumptions

None.

Grading

Grade Components

Component	Weight	Details
Quizzes	**5%**	Students will take weekly quizzes in class. Students will be allowed to drop their lowest 2 quiz scores.
Participation	**5%**	Students will be expected to participate in in weekly Computing in the News forum discussions, posting a new item and responding to others' posts.
Homework	**10%**	Students will complete weekly programming homework assignments.
Written Assignments (Explore Task)	**10%**	Students will complete an "Explore Task" consisting of a paper on on the impact of a computing innovation.
Midterm Project (Create Task 1)	**15%**	Students will complete a midterm team programming project and written report.
Final Project (Create Task 2)	**20%**	Students will complete a final team programming project and written report.
Midterm Test	**15%**	Students will take a midterm assessment in class.
Final Exam	**20%**	Students will take a final assessment in class.

Letter Grades

This Course uses Standard NCSU Letter Grading:

97 ≤ **A+** ≤ 100
93 ≤ **A** < 97
90 ≤ **A-** < 93
87 ≤ **B+** < 90
83 ≤ **B** < 87
80 ≤ **B-** < 83
77 ≤ **C+** < 80
73 ≤ **C** < 77
70 ≤ **C-** < 73
67 ≤ **D+** < 70
63 ≤ **D** < 67
60 ≤ **D-** < 63
0 ≤ **F** < 60

Requirements for Credit-Only (S/U) Grading

In order to receive a grade of S, students are required to take all exams and quizzes, complete all assignments, and earn a grade of C- or better. Conversion from letter grading to credit only (S/U) grading is subject to university deadlines. Refer to the Registration and Records calendar for deadlines related to grading. For more details refer to http://policies.ncsu.edu/regulation/reg-02-20-15.

Policies on Incomplete Grades

If an extended deadline is not authorized by the instructor or department, an unfinished incomplete grade will automatically change to an F after either (a) the end of the next regular semester in which the student is enrolled (not including summer sessions), or (b) the end of 12 months if the student is not enrolled, whichever is shorter. Incompletes that change to F will count as an attempted course on transcripts. The burden of fulfilling an incomplete grade is the responsibility of the student. The university policy on incomplete grades is located at http://policies.ncsu.edu/regulation/reg-02-50-3.

Late Assignments

Quizzes and Computing in the News forum posts will receive no credit if submitted late.

All other assignments will be accepted up to 3 days late with a penalty depending on how late the assignment is turned in:

- at least 1 minute late but less than 24 hours late: the penalty is -10 percent.
- at least 24 hours late but less than 48 hours late: the penalty is -20 percent.
- at least 48 hours late but less than 72 hours late: the penalty is -30 percent.
- at least 72 hours late: the assignment will not be accepted.

Attendance Policy

For complete attendance and excused absence policies, please see http://policies.ncsu.edu/regulation/reg-02-20-03

Attendance Policy

Your full attendance in class if required. From Academic Policy and Regulation REG02.20.3 - Attendance Regulations:

- Full participation in classes, laboratory period and examinations is expected of all students.
- Teachers in 100 and 200-level courses must keep a record of attendance throughout the semester. Instructors may use reasonable academic penalties commensurate with the importance of the work missed because of unexcused absences.
- Valid excuses for anticipated absences must still be cleared with the instructor before the absence.
- Valid excuses for emergency absences must be reported to the instructor as soon as possible, but not more than one week after the return to class.

In the case of an unexcused absence, students will not be allowed to make up in-class quizzes. However, to account for unforseen circumstances, students will be allowed to drop their two lowest quiz grades. Any other work that is missed as a result of an unexcused absence can be submitted late, but it will be graded according to the Late Work policy.

Absences Policy

Academic Policy and Regulation REG02.20.3 - Attendance Regulations lists valid circumstances for an excused absence. Please also note:

a) For non-emergency situations, you must request advance permission to miss a class. If you must miss a class then, in all situations where it is practical, communicate (in person, phone, email) with the instructor before the class to get agreement (examples: you are not feeling well and wish to miss the class; you have a planned trip; you have a court appearance).

b) For emergency situations where you cannot communicate with the instructor in advance, you must contact the instructor as soon as you return to school (within 1 week of returning). Emergency absences must be certified in writing. Emergency illness that was not discussed with the instructor before missing the class must be certified by a physician in writing.

Makeup Work Policy

A make-up quiz or test will only be given under these conditions:

1. You had advance permission from the instructor to miss class as described in point (a) of the Absences Policy above.
2. You have a verified excused absence for an emergency situation as described in point (b) of the Absences Policy above.

Additionally, according to NCSU REG 02.20.03 - Attendance Regulations, item 3.3, "a maximum of four excused absences is allowed per term. Thus a maximum total combination of 4 make-up quizzes plus tests plus assignments will be given. More than 4 make-up quizzes and tests and assignments will only be considered upon written recommendation from the University Counseling Center."

Additional Excuses Policy

None.

Academic Integrity

Academic Integrity

Students are required to comply with the university policy on academic integrity found in the Code of Student Conduct found at http://policies.ncsu.edu/policy/pol-11-35-01

None.

Academic Honesty

See http://policies.ncsu.edu/policy/pol-11-35-01 for a detailed explanation of academic honesty.

None.

Honor Pledge

Your signature on any test or assignment indicates "I have neither given nor received unauthorized aid on this test or assignment."

Electronically-Hosted Course Components

Students may be required to disclose personally identifiable information to other students in the course, via electronic tools like email or web-postings, where relevant to the course.

Examples include online discussions of class topics, and posting of student coursework. All students are expected to respect the privacy of each other by not sharing or using such information outside the course.

Accommodations for Disabilities

Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, students must register with the Disability Resource Office at Holmes Hall, Suite 304, Campus Box 7509, 919-515-7653. For more information on NC State's policy on working with students with disabilities, please see the Academic Accommodations for Students with Disabilities Regulation (REG02.20.01) (https://policies.ncsu.edu/regulation/reg-02-20-01/).

Non-Discrimination Policy

NC State University provides equality of opportunity in education and employment for all students and employees. Accordingly, NC State affirms its commitment to maintain a work environment for all employees and an academic environment for all students that is free from all forms of discrimination. Discrimination based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation is a violation of state and federal law and/or NC State University policy and will not be tolerated. Harassment of any person (either in the form of quid pro quo or creation of a hostile environment) based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation also is a violation of state and federal law and/or NC State University policy and will not be tolerated. Retaliation against any person who complains about discrimination is also prohibited. NC State's policies and regulations covering discrimination, harassment, and retaliation may be accessed at http://policies.ncsu.edu/policy/pol-04-25-05 or http://www.ncsu.edu/equal_op/. Any person who feels that he or she has been the subject of prohibited discrimination, harassment, or retaliation should contact the Office for Equal Opportunity (OEO) at 919-515-3148.

Course Schedule

NOTE: The course schedule is subject to change.

Lecture TH 1:30pm - 2:45pm — Introduction to Programming — TBD - TBD

Lectures: Welcome, Basics of Programming
Readings: None

Programming Lab ("Kaleidoscope"): Get to know the Snap programming environment by creating a kaleidoscope program using abstraction and sequential programming.

Lecture TH 1:30pm - 2:45pm — Functions & Variables — TBD - TBD

Lectures: Welcome, Basics of Programming
Readings: None

Programming Lab ("Kaleidoscope"): Get to know the Snap programming environment by creating a kaleidoscope program using abstraction and sequential programming.

Lecture TH 1:30pm - 2:45pm — Conditionals & Logic — TBD - TBD

Lectures: Conditionals, Logical Operators
Readings: Blown to Bits Ch. 1

Programming Lab ("Guessing Game"): Create a number guessing game that uses conditions to react to user input.

Lecture TH 1:30pm - 2:45pm — Abstraction — TBD - TBD

Lectures: Abstraction, Decomposition
Readings: Blown to Bits Ch. 2

Programming Lab ("Brick Wall"): Create a program that uses abstraction and decomposition to draw a brick wall with adjustable size.

Lecture TH 1:30pm - 2:45pm — Loops & Lists — TBD - TBD

Lectures: Iteration, Lists, Data-structures
Readings: Blown to Bits Ch. 3

Programming Lab ("Tic-tac-toe"): Use lists and loops to create a tic-tac-toe game that stores the game-state using a data-structure.

Lecture TH 1:30pm - 2:45pm — Problem solving with computing (I) — TBD - TBD

Lectures: The Creative Design Process
Readings: None

"Create Task 1": Students will complete a midterm team programming project. They will be given time in class to work on the project.

Lecture TH 1:30pm - 2:45pm — Algorithms — TBD - TBD

Lectures: Algorithms, Complexity
Readings: Blown to Bits Ch. 5

Programming Labs ("Searching" and "Sorting"): Develop algorithms to search and sort lists of numbers and understand the efficiency of these algorithms.

Lecture TH 1:30pm - 2:45pm — Data — TBD - TBD

Lectures: Data & Visualization, Hypothesis Testing, Data-driven Inference, "Big Data"
Readings: Blown to Bits Ch. 6

Programming Labs ("Data Visualization" and "Big Data"): Use programming to visualize and explore large datasets, and compare scientific and data-driven approaches to gaining insight from data.

Lecture TH 1:30pm - 2:45pm — Social Implications of Computing — TBD - TBD

Lectures: Privacy and the Internet, Computing in Education, Saving the World with Computing
Readings: Blown to Bits Ch. 7

"Explore Task": Students will complete an consisting of a paper on on the impact of a computing innovation. Class time will be spent discussing ways in which computing shapes the world and intersects with other disciplines.

Lecture TH 1:30pm - 2:45pm — Recursion — TBD - TBD

Lectures: Recursion, Advanced Programming Topics
Readings: Blown to Bits Ch. 8

Programming Lab ("Fractal Tree"): Students will use recursion to draw a "fractal tree".

Lecture TH 1:30pm - 2:45pm — Problem solving with computing (II) — TBD - TBD

Lectures: Software Design
Readings: None

"Create Task 2": Students will complete a comprehensive final team programming project with a socially-relevant application. They will be given class time to work on the project.

Lecture TH 1:30pm - 2:45pm — Exploring other computing environments — TBD - TBD

Lectures: Transitioning from Blocks to Text
Readings: None

Programming Tasks: Students will be introduced to programming tools beyond Snap, including textual programming and applications such as game making.