# CS 111 Course Syllabus

|  |  |
| --- | --- |
| **Prefix:** | CS 111 |
| **Title:** | Computer Science Foundations 1 |
| **# Credits:** | 4 |
| **Semester/Year:** | Fall 2018 |
| **Pre-requisites:** | MATH 101, MATH 101i, MATH 102 |
| **Meeting Times:** | **Lecture:**  Tu, Th 1-2:20 FH 163  **Lab:**  (11) F: 11:00 AM - 12:50 PM (BSS 313)  (12) F: 9:00 AM – 10:50AM (BSS 313) |
| **Course website:** | Canvas (canvas.humboldt.edu)  Github (see Canvas for link) |

## Instructor Information

|  |  |
| --- | --- |
| **Name:** | Adam Carter |
| **Office location:** | BSS 324 |
| **Office hours:** | M: 10:30-12; Tu: 4:30-5:30; Th 4:30-5:30; F: 2-3:30; Also by appointment |
| **Contact information:** | email: adam.carter@humboldt.edu |

## TA

|  |  |
| --- | --- |
| **Name:** | None! |
| **Office location:** |  |
| **Office hours:** |  |
| **Contact information:** |  |

## Textbooks

Halterman – Fundamentals of C++ Programming (available **free** on github website)

TutorialsPoint C++ / JavaScript tutorials (link on github website and also on course calendar)

## Other Materials

The lab computers in BSS 313 have been outfitted with all the software required to successfully participate in CS 111. However, it would be a major benefit if you outfitted your personal computer with the following programming software:

For JavaScript programming:

* Google Chrome
* Visual Studio Code

For C++ Programming

* **For Computers with Windows:** Visual Studio 2017 Community Edition (Free; Windows Only)
* **For Computers with MacOS:** CLion (Free student edition) ***or*** NetBeans IDE with C++ extensions

## Course Outline

CS 111 is an introductory programming covering problem decomposition, control structures, simple data structures, testing, and documentation. Students design and implement a number of programs. In this course, you will begin to explore the art and science of problem solving using the computer as a tool. Course topics will include problem solving, simple expressions and compound expressions, types of data, syntax and semantics, function application, design, and definition, introduction to software testing, Boolean operations, Boolean functions, conditional expressions, input-process-output, and the basic structures of computing: sequence, conditional, iteration, and procedure.

The problem solving techniques learned in this course will lead to a study of object-oriented programming in CS 112 - Computer Science Foundations 2.

Students are expected to come into this course already comfortable with basic computer use. No prior programming knowledge is assumed or required. I this particular offering of this course, we will alternate between the JavaScript and C++ programming languages. Topics will include discussion of primitive data structures such as arrays and linked lists, higher-level data structures such as stacks, queues, trees, graphs, and hash tables, and concepts such as big-O notation and recursion.

This course lays the foundation for the deeper discussions of algorithms, more advanced data structures, and algorithmic analysis in CS 212 - Algorithms.

## Student Learning Outcomes and Assessment

This course addresses departmental learning outcomes of:

• Computational Thinking

• Technical Writing

• Communicating and Collaborating

This course addresses computational thinking at an introductory level, introducing fundamental computing concepts. It addresses technical writing at an introductory level via program documentation and coding standards that stress reusable code, and it addresses communicating and collaborating at an introductory level via experience pair-programming in course lab sessions. Furthermore, after successfully completing this course, students should be able to:

* Design, implement, test, and debug programs that use each of the following fundamental programming constructs: basic computation, standard conditional and iterative structures, and the definition of functions.
* Analyze the behavior of simple programs involving fundamental programming constructs.
* Choose appropriate conditional and iterative constructs for a programming task.
* Apply the techniques of structured (functional) decomposition to break a program into smaller pieces -- or, better yet, originally design it using such smaller pieces.
* Describe strategies that are useful in testing and debugging.
* Write clear comments that communicate to the reader what a function expects and what it produces.

# HSU Learning Outcomes that this course addresses:

This course explicitly contributes to students’ acquisition of skills and knowledge relevant to HSU Learning Outcomes:

HSU graduates will have demonstrated:

* Effective communication through written and oral modes.
* Critical and creative thinking skills in acquiring a broad base of knowledge and applying it to complex issues.
* Competence in a major area of study.
* Appreciation for and understanding of an expanded world perspective by engaging respectfully with a diverse range of individuals, communities, and viewpoints.

HSU graduates will be prepared to:

* Succeed in their chosen careers.

## Grades

Grades will be based on the following (weights are in parenthesis):

* Participation (5%)
* Homework Assignments (50%)
* Labs (10%)
* Midterm exam (10%)
* Final exam (25%)

Letter grades will be assigned according to the following scale:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 93.00 – 100.00 | A |  | 71.00 – 75.99 | C |
| 90.00 – 92.99 | A- |  | 69.00 – 70.99 | C- |
| 87.00 – 89.99 | B+ |  | 60.00 – 68.99 | D |
| 82.00 – 86.99 | B |  | 00.00 – 59.99 | F |
| 78.00 – 81.99 | B- |  |  |  |
| 76.00 – 77.99 | C+ |  |  |  |

### Homework Policy

All homework is to be turned in on the due date. No late submissions will be accepted.

### Participation

Attendance in class is mandatory and will be collected throughout the semester. Periodically throughout the semester, I will require you to answer questions posed on the course website. Additionally, certain in-class activities may be awarded participation points.

### Lab Policy

You are allowed to miss one lab without penalty to your grade. Otherwise, the following rules apply:

1. Each lab, you are awarded 5 points for attendance and 5 points for completing the lab.
2. In order to receive the attendance points, you must either attend the entire lab. However, you may be allowed to leave early if you complete the lab before the end of class.
3. In order to receive the completion points, you must complete the lab activity by the end of the semester.
4. Students with valid excuses (e.g. car accident, sick, etc.) are exempt from these rules, but must provide sufficient evidence.  What constitutes as sufficient evidence is left to the discretion of the instructor.

### A Note on Cheating

The digital nature of computer science can make cheating a very tempting option. Remember that cheating can result in a failing grade, and perhaps more importantly, ***result in academic sanctions***. In order to make sure that this doesn’t happen, I've included some potential scenarios to help guide your behavior.

1. You and some friends are working together on the next homework and you just finished the assignment. Unfortunately, you have to head off to work. A friend who isn't finished asks to get a copy of the assignment from you so that he can use it as a guide. Wanting to be a good friend, you email him a copy of your assignment on the guarantee that he won't copy. ***This is cheating***. You are not allowed to give copies (analog or digital) of your work. More often than not, your friend will copy and paste some of your work into his assignment. This will likely show up when I run my cheat detection software. ***This is a very common way to get in trouble. Don’t do it!***
2. You and some friends are getting together after class to discuss the assignment due tomorrow. You're done and want to help your friends out. To do so, you discuss the assignment using high-level descriptions (e.g., "I have class X that does Y, which helps me get task Z done") and even draw some nice pictures and/or flowcharts for them to follow. ***This*** ***is not*** cheating and is actually encouraged.
3. As part of a lab, you're required to implement a poker game. A quick Google search reveals several existing implementations. You download the project and modify it to suit the requirements of the lab. ***This is cheating.***
4. Your friend is almost done with his homework assignment but has run into a nasty debugging issue. You take a look at his program and quickly realize that he isn't correctly deleting a dynamically created pointer. You show him how to fix is error. ***This is not*** cheating and is actually encouraged!
5. A student has asked for code help on the course's online discussion forum. You respond with a partial code snippet. ***This is not*** cheating and is actually encouraged!
6. During an exam or other testing environment, you use your phone to ask someone else a question. Alternatively, you receive an email or text message from someone in class asking for help and you respond. ***Both cases are considered to be cheating.*** Communicating in any manner during a testing environment is not allowed.
7. You and a friend are working together on an assignment using the same computer. As a result, both of you turn in the same or nearly identical assignments. As I cannot determine who actually did the work, I must consider this ***to be cheating***. Remember, you must turn in original work typed out entirely by yourself.
8. A friend of yours has taken the class in a prior semester. You ask him for all of prior homeworks and labs, which you use to aid you in completing your assignments. ***This is cheating***.

If you find yourself in a situation that is not listed above and are worried about whether or not it might be considered cheating, error on the side of caution and don't do it. Instead, email me (I usually respond fairly quickly). I'll give you a definitive answer and add it to the list above.

# Students with Disabilities:

If you have any type of disability that may hamper your full participation in course activities, it is your responsibility to inform me of your need for accommodations as soon as possible. I expect to hear from you **within the first two weeks of the semester** so that appropriate accommodations can be arranged.

Complete information on the services available at HSU can be found at the **Student Disability Resource Center** in the basement of the Library, Library 055, **826-4678 (voice)** or **826-5392 (TDD)**. You can reach the Student Disability Resource Center's web site at:

http://www.humboldt.edu/disability/

Please note that some accommodations may take up to several weeks to arrange.

If you qualify for extra time on exams or need other accommodations, it is your responsibility to obtain and provide me with the Exam Accommodation Request form from the SDRC. The form must be presented to me in a timely manner so appropriate arrangements can be made in advance for all exams. I strongly recommend submitting the form at the beginning of the semester or at least one week before the first exam that you wish to use accommodations. It is also strongly recommended that you communicate eligible accommodations and scheduling arrangements with me one week prior to each exam. This will ensure your test accommodation arrangements are completed in a timely manner.

# Add/Drop Policy:

Students are responsible for knowing the University policy, procedures, and schedule for dropping or adding classes. You can find these on the web at:

http://www.humboldt.edu/registrar/students/regulations/schedadjust.html

You can find the University policies for repeating classes at:

http://www.humboldt.edu/registrar/students/regulations/repeat.html

Note that the CSU (and thus HSU) policies on withdrawing from and repeating courses changed as of Fall 2009:

* Students may withdraw from no more than 18 semester-units after the first four weeks of instruction; that is, students may withdraw from no more than 18 semester-units between census and the final 20% of instruction, and only then with a serious and compelling reason.
* Students may repeat courses only if they earned grades lower than a C.
* Students may repeat up to 16 semester-units with grade forgiveness.
* Students may repeat up to an additional 12 semester-units with grades averaged.

**Be careful** – as of Fall 2009, HSU is being much more strict about what constitutes a “serious and compelling reason”.

If you do drop the course, note that it is **your responsibility** to complete and submit the appropriate paperwork. As noted in the University policies for dropping courses, "As a matter of university policy, **the instructor in the course may not drop on behalf of the student**."

# Incompletes:

Incompletes are rarely given and only in the case of a true emergency. They certainly are not appropriate for students who find they have fallen behind on assignments, missed a test, or taken on too much academic, work, or family responsibilities. For these situations, dropping the course would be appropriate (if that is still possible according to the University policies for dropping courses).

# Time Expectations:

Remember the general rule of thumb for college-level courses --- to be successful in a course, you should plan to spend at least 3 hours outside of class for each 1 hour of college course credit. That implies an estimate of at least 12 hours a week spent outside of class for this 4-credit course.

# Academic Honesty:

Students are responsible for knowing policy regarding academic honesty:

http://www.humboldt.edu/studentrights/academic\_honesty.php

Observe that among the actions that are unacceptable are submitting another's program, code, or file as your own and failing to quote material taken from another person’s written work. (Note that copying another student's comments is also unacceptable.)

# Additional University Information:

* **Campus Policies:** HSU maintains a complete list of campus policies at the following website:  
  http://www2.humboldt.edu/academicprograms/syllabus-addendum-campus-resources-policies
* **Attendance and disruptive behavior:** Students are responsible for knowing policy regarding attendance and disruptive behavior:   
  http://www.humboldt.edu/studentrights/attendance\_behavior.php
* **Late arrival to class:** Please attempt to come to class on time, with your headphones put away and your cell phones turned off. If you must arrive late or leave early, please do so with the least possible distraction to other students. If your late/early habits become disruptive, you may be asked to leave the class permanently.
* **Class disruption:** University policy requires that instructors eliminate disruptions to the educational process. Distractions such as excess talking, ringing cell phones, working on assignments for other classes, inappropriate or distracting laptop/tablet/smartphone/gadget use, demonstrations of affection, packing of books early, loud music leaking from headphones, chronic late arrivals or early departures, excessive comings and goings or other behaviors that disrupt the class are not acceptable. Students indulging in such behaviors will first be warned before being required to leave the class permanently.
* **Emergency Evacuation:** Please review the evacuation plan for the classroom (posted on the orange signs), and review the campus Emergency Preparedness web site at: http://www.humboldt.edu/emergencymgmtprogram/campus\_emergency\_preparedness.php   
  for information on campus Emergency Procedures. During an emergency, information regarding campus conditions can be found at **826-INFO** or:  
  http://www.humboldt.edu/emergency