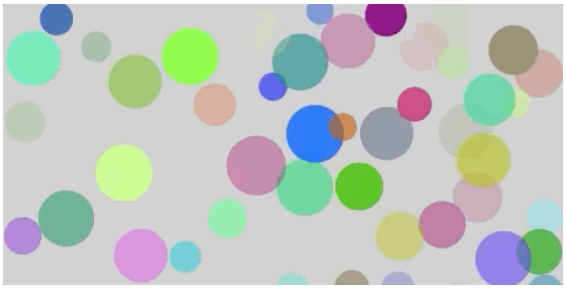
Computer Programming in Java - Fall 2021/Spring 2022

Syllabus

Teacher Chandru Narayan

Room **Gracemont 204**

E-mail chandru.narayan@bush.edu



Course Description

This course is designed to introduce computer programming in the Java language. Learning to use a computer language is a necessary skill for all students regardless of discipline. In this course we will teach the fundamentals of computer programming from the stand point of simulation, automation, and problem solving of real-world systems and natural processes. At the same time, the design and implementation of computer programs is taught from the context of fundamental aspects of computer science, including the development and analysis of algorithms, the development and use of fundamental data structures, the study of standard algorithms and typical applications, and the use of logic and formal methods.

In addition, the year-long course will cover many of the topics necessary for preparation to the AP Computer Science A examination in Spring of the following year. This is an introductory course in computer programming using Java. As such, no specific programming prerequisites are needed to take this course. However, additional work may be needed to fully prepare a student for the AP CSA exam with no prior knowledge of computer programming.

Desired Outcomes

Students should be able to at the end of the year-long course:

- Design and implement solutions to problems by writing, running, and debugging computer programs.
- Code fluently in an object-oriented paradigm using the programming language Java.
- Use and implement commonly used algorithms and data structures.
- Read and understand a structured program consisting of several classes and interacting objects.
- Recognize the ethical and social implications of computer use.

21st Century Learning Expectations

Students will create web applications (solutions) not just limited to learning the Java language

- Visual and Project-based learning producing tangible results
- Publish code to Github accessible over the web
- Learn to simulate natural processes and systems through code
- Develop web-based computer games like Super Mario, Asteroids
- Learn and code advanced Math and Physics concepts
- Develop computer web-based dynamic digital artwork
- Foundations for Evolutionary code, Machine learning and Artificial Intelligence

Students will develop basic computing skills regardless of their discipline of study:

- Engaging in inquiry based, self-driven, authentic learning
- Utilizing higher order thinking skills that include the evaluation and synthesis of information and ideas
- Collaborating and communicating with others while also self-reflecting on personal growth
- Applying appropriate technology and media literacy skills
- Making connections to life and the larger world community

Course Materials

Tools/Software:

- Bush Portal
- CPJava Google Classroom
- Bush CPJava course website
- Github & HTML
- Codingbat and trinket.io online exercises
- Processing IDE or Visual Studio Code IDE
- CPJava Slack Channel

Texts/Online Resources:

- APCS A Curriculum & Reference - CSAwesome
- The Coding Train Youtube Channel

Computing:

- Laptop capable of running local applications
- Robust wireless connection when learning Remotely

Classroom Expectations

- Bring your laptop to class everyday. Make sure that it is fully charged.
- Arrive to class on time and be ready to begin immediately.
- Absences:
    - If you have a planned absence from school (such as commons duty, sport's game, or doctor's appointment), you should inform me before the absence.
    - For any absence (planned or unplanned), you are responsible for checking the portal and asking a classmate about missed information ideally prior to the next class meeting.
- This class will have you in charge of managing a lot of your own work. You need to be an independent worker. All schedules and assignments will be clearly posted in the Portal with links to Google classroom/CPJava website.
- Take care of other business outside of the classroom and expect to be working for the full period. You will be working on your laptop regularly and with your classmates. Class time is not a time to check e-mail, peruse social media, etc. Don't cheat yourself of valuable class time to ask questions and work with your peers.
- Working together is a great way to learn computer programming. However, if you find that working with peers is less like collaboration and more like appropriation, please talk to me so that we can make a plan so that you can do your best work.

Typical Student Workflow

1. Attend all classes in-person unless directed otherwise
2. Each Unit will have a presentation that we will follow which is available on the CPJava website
3. Each Unit will also have Exercises and Projects
4. Exercises are short code extracts, videos to watch etc. Projects are complete solutions (programs) that you will write to meet a given specification.
5. Complete these and submit them to Github these as directed. Projects will always have a code submission that is due. In some cases exercises may not have a code submission, but you will still practice them online.
6. Once you complete the exercises and projects, mark them as complete in the Google classroom. If you actually submitted code to Github, paste a link in the Google classroom assignment before marking it as complete.

Grades

Projects and weekly exercises: The majority of class time will be spent completing problem sets to practice and reinforce a computing mindset. We will apply the computing skills we gain to multiple Projects to analyze, document and publish the results. A small set of modular programming tasks with accompanying online exercises will be due approximately once a week. It is your responsibility to use class time to successfully complete and submit code solutions. You have two additional class periods after the problem set is submitted with comments to make corrections and earn full credit. I will be available during conference hours and at other times by prior appointment to offer any help.

Exercises/Assignments **35%**
These will typically include worksheets, and online assignments to complete. These will be submitted to the Google Classroom or completed online directly.

Projects **50%**
There are no quizzes or final exam for this course. The bulk of the grade is earned through completing a dozen or or so projects throughout the year submitted to Github. The projects will be evaluated on the basis of whether or not your program meets the documented specifications.

Student Portfolio Management **15%**
This category is assessed by evaluating the timely submission of projects to the Github portfolio. The student portfolio should be developed such that each project will have a operable link at the top-level of your portfolio website.

Syllabus & Schedule

LESSON UNITS	APPROXIMATE DURATION	TOPICS
FALL 2021 SEMESTER	**12-Weeks**	**Sep to Dec 2021**
Unit 1	2-Weeks	Introduction to Java, Tools walkthrough, Classroom processes, Environment setup, Java Primitive types and Operators
Unit 2	2-Weeks	Objects, Methods, String, Pointers, Integer, Double, Math
Unit 3	2-Weeks	Control flow, Booleans, If statements, Object traversals, Integer, Double, Math
Unit 4	3-Weeks	Iteration, While loops, For loops, Nested Loops, Loop Analysis
Unit 5	3-Weeks	Anatomy of a class, Constructor, Accessor, Mutator Methods, this keyword
SPRING 2022 SEMESTER	**15-Weeks**	**Jan to May 2022**
Unit 6	2-Weeks	Arrays, Array Traversal, Data Structures, Project work
Unit 7	3-Weeks	Array List, Big data, Ethics of Data Collection, Privacy
Unit 8	2-Weeks	2D Arrays, Traversal, Table representation
Unit 9	3-Weeks	Inheritance, Encapsulation, Hierarchy, Polymorphism, Multi-part Project
Unit 10	3-Weeks	Recursion, Recursive Search. Recursive Sort
Unit 11	2-Weeks	Final Project Peer Sharing Final Project Presentation or APCSA Exam

Attribution

This course draws from the materials used in the the CSAwesome/TEALS curriculum as well as CS programs at Lowell High School in San Francisco and Boston Latin School. Significant credit is also due to the many member founders of the Processing Foundation, especially Dan Shiffman whose tireless efforts have advanced the cause for open, accessible, and free CS education across the world.