# Introduction to Java Programming

Mr. McLeod (Room 723), Ms. Woo (Room 312)

**Cupertino High School 2017-2018**

## Course Description

Welcome to Introduction to Java Programming! The goal of this course is to learn programming in a fun and interactive setting. The course will begin with media-based coding tools such as Scratch, codehs.com, and Greenfoot and transition to pure Java by the second semester. This course will give students an opportunity to build their problem solving skills.

**Course Website:** www.tinocs.com

## Grades

Grades are weighted as follows: Labs & Homework 50% Tests/Quizzes 50%

| | | | | |
|---|---|---|---|---|
| A ≥ 90% | B ≥ 80% | C ≥ 70% | D ≥ 60% | F < 60% |

## Grading Philosophy and Tests

The grading structure balances student proficiency, progress and effort level. Most homework is graded purely on effort. If you put in genuine and honest effort in a timely fashion you will usually get full credit. However, you can't submit work late or be completely lost and not understand anything. Assuming 100% HW, here's the minimum requirement to get different grades:

| HW | Test% | Overall Grade |
|---|---|---|
| 100% | 80% | 90% |
| 100% | 60% | 80% |
| 100% | 40% | 70% |

A lot of the work we do during class is fun and collaborative. However, you should know that you will be individually responsible for understanding what's going on. If you receive help from others and you're unable to complete the same problems on your own, you need to get help immediately. If you do nothing significant to correct this quickly, you will not be able to catch up. Here are some interventions you can use:

- Get help from Ms. Woo or Mr. McLeod during class
- Plan out and start writing your code in Ms. Woo / Mr. McLeod's room at lunch or after school
- Find a current or former APCS / Intro Java student who can work with you in person without referring to their own code
- Use tutorial to debug with a partner or get help from Ms. Woo / Mr. McLeod
- Find a private tutor or go to one of the many Tutoring Clubs in Cupertino

### Make Up Test

If you have any test score below 60%, you may retake a test under the following conditions:

1) Your overall homework grade must be at least 80%.
2) You must work with the teacher to complete a list of assignments designed to help you prepare for the test.
3) There will be specific times when recovery tests are given, so all preparation must be completed before the scheduled test date.
4) The maximum score for the recovery test is 60%.

## Homework Policy

**All assigned homework is due on the due date by midnight unless otherwise specified.** You are encouraged to work with other students throughout this course. Note however, that there is a difference between helping someone and giving them your solution. And there is a difference between learning from someone else's code and using their general ideas versus copying their code. Unless told otherwise, you are expected to come up with ***your own solutions to problems.*** If you are copying code or partial code for an assignment that is meant to be done individually, you are not doing the right thing and you may be cited for a cheating violation. For more details, please see the **Academic Integrity Policy**.

## Labs / Homework Late Policy

You are allowed to turn in late assignments for 50% credit as long as you turn the work in **no later than one week** past the assignment's due date. After that, it's a zero.

## Labs / Homework Resubmission Policy

You are allowed to resubmit assignments for up to full credit that were turned in on time, but the resubmission must be turned in **no later than one week** after the assignment's due date.

## Labs / Homework Extension Policy

For both late work and resubmissions, you may be granted more than one week to complete an assignment only if you fill out an extension form within a week of the due date.

## Supplies

- A computer at home with internet access.
- A way to transfer files to and from home (USB Flash Drive, Google Drive, Dropbox, etc).
- Notebook for notes and warm-ups.
- Headphones to listen to videos/sound.  Your regular iPod earbuds will do just fine.
- Having a laptop with wireless access is useful but not required.

## Tardies/ Absences

It is important for students to attend class regularly and to be on time for class. Students who are absent from classes are expected to take responsibility for understanding the lessons that were covered during their absence. For excused absences you have as many days as you were absent to make up the work. **To make up the work, you and Parents/Guardians need to inform the teacher by email.** You are not allowed to makeup or turn in work for unexcused absences.

## Classroom Rules

The classroom is a learning environment and students should behave accordingly. In general, this means that students should not:

- do anything that actually or potentially disrupts learning
- use cell phones or any form of electronic entertainment in class
- eat or drink anything other than bottled water in the classroom

## Parents

Please feel free to contact us whenever you have questions or concerns. You can e-mail us via School Loop or directly at ted_mcleod@fuhsd.org / hyungi_woo@fuhsd.org.

## Intro to Java Academic Integrity Policy

You are encouraged to help other students in this course. You may help each other make a plan and solve problems using paper and pencil. You may also help another student debug their code to discover an error and give advice for fixing it. If you struggle to complete an assignment or life gets too busy, take advantage of the resubmission policy. Don't give up and turn in your own version of someone else's code. It doesn't pay off. Using someone else's code as part of yours or as a template is considered an academic integrity violation and will get you in a lot of trouble. Also, do not make your code available to other students as you will be held responsible if they use your code even without your knowledge. It's important that students find their own solutions to the problems given. When it comes to writing programs, you are required to write your own code. You may not copy any part of another person's program nor use it as a template for your own program. Also, you may not send your programs to other students except as outlined in the Debugging section. If you do so, you and whomever you copied/received from may be cited for CHS Academic Integrity Violation which can result in removal from the class and can affect your college admissions.

The general guideline is that you may learn from others and teach each other but not at the expense of turning in work that is someone else's or compromising the intended goal of you solving a problem yourself.

### Writing your own code

**BAD** You copy someone else's program and turn it in as your own.

**BAD** You copy someone else's program and but change it to look like you did it yourself.

**BAD** You write your own program but get stuck with one or two parts. Someone else in the class already has a solution to those parts so you copy those parts.

**BAD** You and another student work on an individual assignment together and submit the same code.

**GOOD** You write your own program. It's not the best, but it works.

**GOOD** You write your own program. It's not the best, and it doesn't work. But you put in hard work, solid time, and effective effort. If this happens it's ok. You get help from the teacher and resubmit as per the redemption policy.

**GOOD** You write your own program. It's not the best, but it works. You then review another student's code and see a better approach or technique. Either turn in your ugly code and use the technique you learned next time or, on your own, modify your program using the new technique. Using techniques you have learned is good. Copying and turning in someone else's work as if it were your own is bad.

### Debugging

**BAD** You get stuck and can't find what's wrong with your program so you send it to a classmate to review. Your classmate fixes the problem for you and sends it back to you.

**GOOD** You get stuck and can't find what's wrong with your program so you explain to another classmate what the problem is and what you have tried so far. They tell you what they think might be wrong and offer advice for fixing the problem. They do not send you code that fixes the problem - they just explain how to fix it. You write your own code based on their advice and it works.

**GOOD** You get stuck and can't find what's wrong with your program so you send it to a TA or a classmate who has already finished and submitted the assignment. They find the problem and explain what's wrong and why it went wrong, but doesn't fix the problem for you. You understand the problem and fix it yourself. The classmate removes your file from their computer.

### When You Get Stuck Conceptually

**BAD** You have code written, but you get stuck with a particular part of the program and don't understand how to attack the problem. You ask a classmate about it and they send you their code. You use their code as if it were your own.

**GOOD** You have code written, but you get stuck with a particular part of the program and don't understand how to attack the problem. You ask a classmate about it and they explain it verbally or online. You are able to write the code yourself from the explanation.

***Please sign the agreement on the back.***

**I have read and understand the information on the Syllabus and Academic Integrity Policy.**

**PRINT your name legibily:** **Period: ___________ Date: ___________________**

**Student Name: _________________________________________ Signature: _________________________________**

**Parent Name: __________________________________________ Signature: _________________________________**