What is This Course?

Introduction to Programming with Java is one of DMA's most popular courses. Literally thousands of student take this course each summer, so it's an important one. The goal of the course is to provide an introduction to programming for students using the Java programming language.

Why Java? Java can be hard! This is true, but it's also very transparent, extremely popular, and has a good combination of explicitness and abstraction. In order to make it easier, the course actually starts in a different programming language, Processing.

Processing is an open-source programming language created by some awesome people at MIT, and it's great. It is based on Java, and is in fact a simplified version of Java. It removes the need (but not the functionality) of a lot of the advanced things that Java requires, which can get in the way for a new programmer. Things like public static void main(String[] args), or declaring everything public vs private, or even containing EVERYTHING within a class.

This will allow students to start programming from minute one without having to deal with all those things. Will the course eventually go over those things? Absolutely. But not on Day 1.

Who is Taking This Course?

This course is part of DMA's Teen program, which means you can expect your class to consist of students anywhere between the ages of 12 to 18. This can present a challenging teaching environment, but you can get some tips below in the Teaching Practices section of this page.

The experience of the students in your course will also vary. Some will be completely new to programming, some will have some experience in school, some may even be pretty great at it already (but that should be rare). You will probably encounter students who have gone through the DMA online eLearning course, Fundamentals of Programming. These students will have spent several hours online learning to program with a Javascript version of Processing. You will also encounter students who are in week 2 of DMA's Academy of Programming 101, which means they will have just spent a week learning to program games in python. For both of these groups students (some may even belong to both!), day 1 may also be a little repetitive. So again, please check the Teaching Practices section.

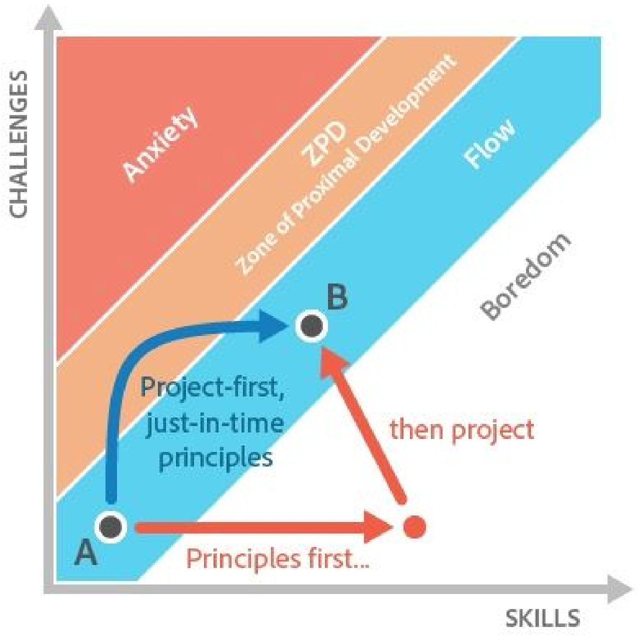
How to approach this curriculum?

Yes. You may know Java, but the lessons here in this course will provide you with a great way to teach Java, and more importantly, the core programming concepts. This course is also a part of several academies, which means that students will be moving on from this course into one of several other more advanced courses, and the curriculum in those courses will probably build on what you taught in this course. Also, there are probably things in this course that you are not familiar with, such as Processing, or how to using processing as a graphics library in Eclipse.

Does this mean you shouldn't experiment, show the students some your own projects, or even come up with your own lessons? Of course not! You are encouraged to do so. Just be sure you are covering the concepts outlined in this curriculum. And if you do create original stuff, share it with your fellow instructors!

DMA believes in the learn by doing model in which a student works through the design process in a creative, positive and supportive environment. Project based learning activities are the backbone to a successful DMA experience. Students will be engaged in hands-on learning experiences as they are challenged to address high level questions and develop solutions to problems through an iterative process.

Project Based Learning Model



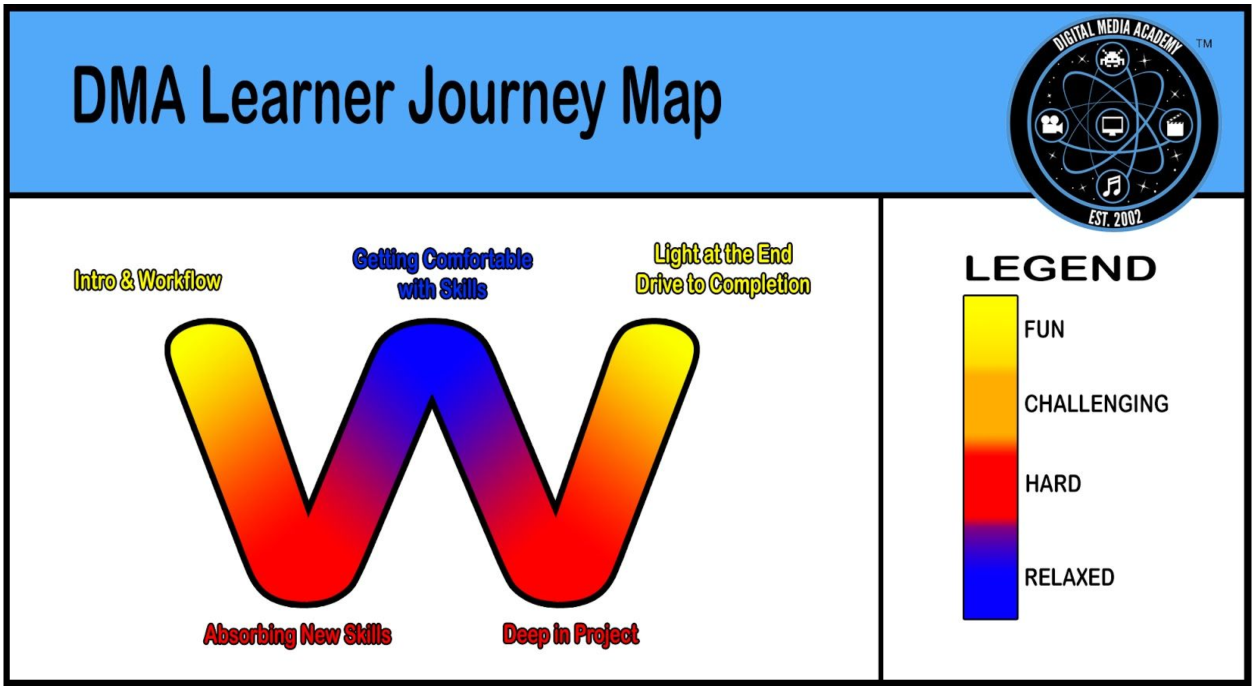
WHAT TO AVOID:

When concepts are taught first without a context for application and then followed by a project in which they are applied, there is a higher likelihood of student boredom or detachment.

WHAT TO DO:

However, when an inquiry- or project-based lesson is presented first, the just-in-time scaffolding of underlying concepts enables students to reach the zone of proximal development and a state of mental flow more quickly. In other words, students are more engaged and stay focused longer.

During a week of instruction a student will experience a range of emotions. Paying attention to how a class or even a project is structured to lead a student through this range is a fundamental step in the development process to ensure a student’s success. This design is intended to provide a balance of fun, challenging, difficult and relaxed moments at the right times to keep a student engaged and motivated.



Best Practices - The DMA Model

The DMA model of watch me, do with me, and then do on your own holds very true to programming. Have students follow along with you when presenting a subject, and then allows them to practice those skills. For example, write a for loop, explain it, then have the students write another for loop following you, while re-explaining the concepts, and then give them small challenges and exercises to practice for loops.

Dealing with Advanced Students

For both advanced and struggling students, TAs are really important. Ask your TAs to challenge advanced students by giving them extra challenges that you can find in the advanced extensions of the lessons. Make sure the TAs know who those students are. If you have a large a group of advanced students, you can even assign a TA to watch over that group specifically, and even do advanced lessons (with advisement from the instructor).

Dealing with Struggling Students

Be sure to identify students who are having a hard time on Day 1. Go out of your way to engage them and make things clear. Again, point them out to your TA. You can also try to seat them next to an advanced student who doesn't mind being helpful. Use checkpoint lessons and work time to sit down with the students (or have a TA do it) to help bring them up to speed. It's important to manage your time to make this possible. Don't try to move the class to fast if a group of advanced students wants to keep going. Instead, provide them with extra challenges or tasks.

Dealing with Troublesome Students

Most of the students you meet at DMA are pretty great. Sometimes you get students who have behavioral problems. Most of the time, it is simply a matter of getting these students engaged. Try to give them assignments or even responsibilities instead of punishment. If that is not effective, simply a few warnings that the student will have to go without computer time can be helpful to encourage good behavior. Beyond that, simply notify your Staff Director and they can contact the parent for you.

Dealing with Special Needs

Since our courses deal with pretty advanced subjects, most of the special needs students you will see will be high functioning. Usually, this means they fall somewhere on an autism spectrum.Read this (Links to an external site.) guide on dealing with autistic students. Beyond that, you can really just treat them like other students, or more importantly, treat all students the same. It is NOT your position to diagnose. This is what Autistic students will respond to the best. It is NOT your position to diagnose. Parents should provide any necessary information which should show up on your roster. If you need to do something special, note that and do you best to accommodate. It is important to remain patient. Beyond that, remember that EVERY student is different :)

Syllabus

Part 1: Introduction to Programming using Processing

- Introduction to Processing, Why learn Processing
- The First Program: Hello World, Your First Drawing
- Using Math
- More Shapes: Rectangles and Ellipses
- Draw a Face
- Fix the Broken Traffic Light
- Color Theory
- Bonus Shapes - Triangles and Optionally, Arcs
- Be Creative - Draw a Creature and add details
- Advanced Extensions

Part 2: More Processing (Handling events, functions, etc)

- Mouse and Keyboard - Dynamic Mode, mouseX, mouseY
- Creating Functions - setup() and draw
- Variables - Greater Detail
- Conditionals - if, else with mousePressed
- Keyboard - the AND and OR operators

Part 3: Using Processing with eclipse

- Intro to eclipse
- Creating a Processing Project
- The differences between Java and Processing
- Creating a Function

Part 4: Variables

- Quick Challenge: draw a rectangle
- Add global variables
- Update Variables
- Compartmentalize the code

Part 5: Object Oriented Programming

- What is an Object
- What is a Class
- Modeling an Object

- Writing a cookie cutter
- Using an Object

Part 6: Starting a new project

- Create a new Project
- Create a new Class
- Use Color
- Create Accessors
- Create Modifiers
- Use Color Correctly

Part 7: Arrays

- Learn about what an array is
- Update our ball program to use an array
- Update it again to use ArrayLists

Part 8: Putting it all together

- Create Project, Import Files
- Add the Ball to the screen
- Add the Paddle to the screen
- Add the Bricks to the screen
- Implement bouncing off the paddle
- Implement bouncing off the bricks
- Lose Condition
- Advanced: (Debug)