48-257
51-257
51-757
60-257

Programming for the Arts with Processing

Jim Roberts, jr2u@andrew.cmu.edu

--

http://www.andrew.cmu.edu/course/60-257/

Spring 2014 - revised

College of Fine Arts
Carnegie Mellon University

Staff

Jim Roberts, e-mail: jr2u@andrew.cmu.edu
TBA

Description

Programming for the Arts with Processing (a/k/a 257) is a first course in programming for students in the College of Fine Arts. Even though it is designed for students with very little or no prior programming experience, the course is not a "slow" course or an "easy" course. The course operates on the idea of, "see it, do it, then talk about it," "See it," means that we write code during class to explore new topics. "Do it," means that you write similar code for your own practice and for homework assignments. Talk about it happens when you have questions about your attempts. The material is presented in a stair step pattern with each step forming the foundation for the next step. Mastery of the material at each step is necessary in order to achieve success in the following steps. The class assumes that you know how to use a computer and a mouse. Beyond this, we will provide guidance, a bit of information, and many chances to explore.

This course will expose you to the fundamental constructs of many programming languages (Java, C, C++ to name a few), types, collections of data, procedural programming, object oriented programming. It will also expose you to planning, coding, debugging and testing of programs. Every assignment will require you to write a program that has a graphics output. One of the overall goals is the animation of data in a meaningful manner.

In keeping with university tradition, much of what you learn will be on your own working alone or with others in the class. Like most college courses, we ask more questions than we answer.

Collaboration is expected. You are encouraged to work together. The caveat is that you must understand the code that you are asked to write.

Work Load

You have to participate in the class. Just sitting in and listening will not work. You will see some grades in the second week of class that show the results of just sitting in. Likewise, missing one day in two or going away for two weeks when other stuff gets busy are very flawed strategies. 257 is a nine-unit course. This means that you owe 257 nine hours on average a week. Four of those are class time leaving five outside class – an hour a day for five days a week. Not every week will take nine hours of your time but a few might take twelve or fifteen. You need to work every week. Ideally, you need to work a little every day. Programming is closer to piano than physics.

Copyrights

As part of your assignments, you will either have the opportunity or be required to display images, sound, and video. The web is loaded with this type of data. Much is copyrighted. You can legally link to a web page with an image from a web page or program, but you need permission of the copyright holder to embed the data in your programs when your code displays the image, video, or plays the sound.

Your assignments are in a very public space and visible to the entire world. You must be sure that you honor the idea of copyright when you are seeking images for your programs. You can take pictures yourself, or find public domain images, or contact the owner of the image for permission. This is your responsibility.

If you think no one would look at your work, you are terribly mistaken. I received mail last semester from other schools and individuals commenting on students' work when student work was posted on the web.

Going in the opposite direction, you have to protect your intellectual property. You are required to include a copyright statement in a comment at the top of every program you write. It is just possible that you will dream up the next fortune-making software idea. You never know what will happen when you start writing code. *Jim is beyond his retirement age and he is looking for software ideas he can license to others to pad his retirement funds. If you don't copyright your idea, he might.* Protect yourself.

Textbook

There is one strongly recommended text: *Learning Processing* by Daniel Shiffman. This book has a bright orange cover. Last year Jim used *Processing; Creative Coding and Computational Art* by Ira Greenberg. Shiffman starts with the very basic fundamentals and moves at a very reasonable pace through the material. Greenberg assumes more and moves more quickly. Either will do but Shiffman is recommended. Shiffman is available in the CMU bookstore. Amazon.com sells both. A link to both books on Amazon can be found on the Processing web site. Several of you may want to buy one copy and share it instead of purchasing it for yourself. It is not expensive by most textbook standards – around $50.00. This is close to the bookstore price.

Programming Environment

We will use the Processing IDE (Integrated Development Environment). It is available on the web and is free. It works on Linux, Windows, and Mac – download details will be provided in class on the first Monday. It also comes with boatloads of example code, documentation, and a very well thought out API (Application Programmers Interface) which is sorta' like a dictionary. The developers of Processing, Reas and Fry, are MIT computer scientists by degree but at heart, they along with Shiffman and Greenberg are artists. They developed Processing as an extension of Java but one that is much simpler to use. The documentation or API is written for artists and designers. There is very little geek-speak in their writing.

Classrooms

We meet in 5202 Wean Hall in a newly designed cluster referred to as the CTC (Collaborative Teaching Cluster).

Class Structure

257 meets two days a week for 110 minutes. We will try to follow this pattern for the first part of the course as we work through the basics:

Tuesday	Thursday
Admin Stuff	Admin Stuff
Revisit last week's stuff. Look at a new area or topic	Revisit Tuesday's stuff Look at the short homework Finish looking at the new stuff
Begin work on a short homework due Thursday in class	Begin **work on** a more involved homework due the next Tuesday in class,

Once we get through the fundamentals, we will alter the pattern to fit the size and breadth of the assignments.

Missing Class

Missing class for a CMU approved reason or for health reasons is acceptable and the work may be made up. Missing for other reasons does not entitle you to make up the work. This means that you cannot come for "individual" lectures from any of us for a class that was cut. Obviously, you cannot anticipate illness. Check in with Jim when you get well. ***Don't come if you are sick*** – nothing personal – but we'd like to stay as healthy as possible and not catch whatever you have. Advance notice of absences for CMU approved reasons is appreciated.

Class Lecture Notes

There are no printed lecture notes but a summary of stuff done in class is occasionally posted as a PDF file on the "Notes from the Board in Class" web page. You really do need to come to class.

Homeworks

A homework is a program that you write to meet a set of specifications. The specifications involve the new topics just covered in class. Old topics will also be included since nothing ever goes away. The tentative schedule calls for lots of programs to be written two a week for the beginning part of the class. This will change as we move through the basics.

Homeworks must be submitted without syntax errors – they must compile. Homeworks that do not compile receive a grade of zero. A homework program that crashes or fails to meet all of the specifications will be marked down. Handin instructions will be given in class during week 1 and 2.

Homework Grading

100% is max grade for handing work in before the deadline. Work submitted after the dedline can be submitted up to the next class exam for a max grade of 90%. Work submitted after the next class exam will receive a grade of zero.

Portfolios

You must maintain a portfolio of work on Processing.org. You need to access this web site and register for a space. The link is:
www.openprocessing.org

Bonus Work

Bonus points for "extra" work may be announced in class but is never posted on the web. This is a good reason to come to class.

In Class Exercises (ICE)

In Class Exercises may be assigned on any class day. These are designed to get you ready to think about the current homework. These are assigned in class and due before the class is over. There is no make up for ICEs. They are graded as pass/fail If the assignment is not finished, is not done properly or you are absent, the grade is fail. The grades on the ICEs are used for final grade borderline cases. They will have no direct negative impact on your final grade but in cases where you are "close" to the next letter grade, they could result in the grade being rounded up.

Course Exams

There will be three on-line exams – on Thursday in weeks 5, 10 and 15. These three exams are each worth 20% of the overall course grade. In total they count for 60% of your overall grade. Exams will have coding questions where you have to write code to solve a problem. The API will be available for the exam but you will not be allowed to refer to code written by you or others during the exam. Exams from prior terms will be posted on the calendar web page.

No exam grades are dropped. Every exam counts. The exams can be difficult if you do not code regularly and keep up with the work. The best way to prepare for the exam is to write code every day.

Projects

You will write two projects in the final part of the course. These will be individual, self-defined projects within a provided context.

Final Exam

There is NO final exam

Late Work

Late homeworks are a problem. If we give you no credit, we risk your not doing the work. Each homework has set of specifications that require that you to write a program. Ideally, in the process of developing the program and writing the code, you will explore and begin to understand the concepts required to meet the specifications. Consequently, it is vital that you do and understand the homework in a timely manner. To encourage you to do the homework, there is only a minor 10% penalty for late work. Homework is designed to prepare you for the exam. Therefore, all homeworks that are covered by Exam 1 must be submitted prior Exam 1. Homeworks not submitted prior to Exam 1 receive zeros for grades. The same applies to Exam 2 and Exam 3. There is no bonus for turning work in ahead of the deadline.

Extended Time

If you are due extended time for exams, please provide Jim with the appropriate paperwork during the first two weeks of class and remind him the week before an exam so he can make arrangements. Remaining in the classroom is usually not an option. The rooms are used every hour and are full so you cannot stay late to work. See Jim during office hours to discuss this. Extended time is not given for homework assignments because they are assigned over a longer period of time.

Religious Obligation

Given the diverse nature of the student body at Carnegie Mellon, it is difficult to plan a class calendar and miss all days of religious importance. If an assignment or exam falls on, immediately before, or immediately after a day of religious observance, please talk to Jim in advance so an alternate date can be arranged. There are no grade penalties whatsoever for these arrangements.

Assignment Submission

This will be explained in class in week 2.

Grade Corrections and Changes

You can meet with Jim to discuss a grade on any assignment or exam for a period of one week after the grade is posted.

Computing Your Overall Course Grade

The following table shows how your final grade will be determined:

Course Component	% Weight of Final Grade
Exam 1	20 %
Exam 2	20 %
Exam 3	20 %
All Homeworks	20 %
Project 1	10 %
Project 2	10 %
Total Points	**100 %**

The following grading scale will be used for the class:

Your Final Point Total	Your Final Letter Grade
90-100	A
80-89.999	B
70-79.999	C
60-69.999	D
Less than 60	R

Calendar

The calendar changes very frequently. Jim tries to keep it accurate as far as the past is concerned. You should check it frequently at this link:

`http://www.andrew.cmu.edu/course/60-257/`

Important changes will be announced by e-mail from either Jim or Gilbert. We will try to keep the level of mail small but things happen.

The Words of Others...

The following has been taken from syllabi developed by other instructors from other introductory classes in the past. It is provided for your consideration. More along this line from Jim in class… *Your success in this course will mainly depend on your ability to keep current with material and projects. All programming courses are very time consuming. In addition to attending classes, you must be prepared to spend at leas more time on reading and programming assignments. One of the common mistakes students make in this class is waiting until the last minute to start a programming assignment. Although many other course projects can be handled that way, programming projects are different. You are almost certain to run into three types of problems, technical, algorithmic and debugging. Technical problems frequently occur with computers you use. Computers often break. Networks go down. Therefore waiting until last minute may be hazardous to your academic health. Secondly, most program algorithms are not trivial. They need thought and may require some back and forth discussions with your instructor, course assistants and your classmates. Even if there are no technical or algorithmic problems, there is no guarantee that your program will work. Students frequently run into syntax and logic problems in a program. Error messages are not very good. Programs don't always necessarily give the correct output. Making a program almost "perfect" can require hours of testing. THEREFORE, START AND FINISH YOUR PROGRAMS AND PROJECTS EARLY*.

Cheating

The simple form of this part is, "Do Not Cheat." Exams must be done by you with no outside help of any form using only the API. Information from other students or sources not listed by the exam is not allowed. Use of non-allowed information is cheating and can result in a reduced final grade, failure in the course, and expulsion from the university. Jim is required by University policy to notify your Dean when such a cheating incident occurs. He is also required by University policy to record a grade that is less than the grade given to a student who does not take the exam. Translated, this means a negative grade must be recorded. Additional sanctions by a student's home department, school and Dean as well as the university's Dean of Student Affairs are at the discretion of those individuals.

Homeworks can be done in collaboration with others in the class as long as you list your collaborators in the comment section of any code that was developed and write your own code. Please note that your homeworks contribute 10% to your overall grade. Relying heavily on other students to develop the programs will not insure a passing grade. 60% of your grade comes from exams done in class by yourself. The two projects worth 20% must be done individually and must be different from other projects. You have to learn how to program.

Each term several students come to office hours to argue that their grade is unfair because they have all 100%s on the homeworks yet they have failing grades on exams. This is usually because they relied on others for much of the effort and work on the homeworks and projects.

Do not fall into this trap.

With all of that said, welcome to 257. I have looked forward to teaching a class like this for 30 years. I learned a long time ago that if you are not having some degree of enjoyment if not outright fun, you are not learning. I also learned that if I am not having fun watching you have fun, my teaching stinks. I hope that you will learn and that my teaching will not smell too badly. Thank you for entrusting me with this exposure to programming. I hope this is a great semester for all of us.

By the way. . . My name is Jim. Dr. Roberts is my wife. Professor Roberts was recently fired by SCS, Mr. Roberts dies at the end of the movie[1]. My name is Jim. If you are uncomfortable with that, then find a way to get comfortable.

As for your names. Sorry but leaning a new programming language or a new operating system is easy. The learning the name of a new student is almost impossible (some have suggested age is the cause but . . .). I will sincerely try to learn all of your names but if I call you Lynn and your name is Terry, just correct me.

Once again, welcome to the class.

Jim

[1] **Mr. Roberts is the title of an Oscar winning move released in 1955 starring Henry Fonda as Lt. JG Doug Roberts (Navy) serving in the Pacific Theater in World War II. The movie was directed by John Ford.**