# P0: Final Project

* Your final lab project may be completed individually or in a pair as pair programmers.
* Your task in this final lab project is to expand code created earlier in the term in a direction of your own choosing and using an appropriate object-oriented class structure.

## Learning Objectives

* Synthesize element of learning from the entire course into a single, final project

## Background

So far in this course, you’ve learned about the following programming principles:

* [Chapter 1](http://openbookproject.net/thinkcs/python/english3e/way_of_the_program.html) *The way of the program*
* [Chapter 2](http://openbookproject.net/thinkcs/python/english3e/variables_expressions_statements.html) *Variables, expressions, and statements*
* [Chapter 3](http://openbookproject.net/thinkcs/python/english3e/hello_little_turtles.html) *Hello, little turtles!*
* [Chapter 4](http://openbookproject.net/thinkcs/python/english3e/functions.html) *Functions*
* [Chapter 5](http://openbookproject.net/thinkcs/python/english3e/conditionals.html) *Conditionals*
* [Chapter 6](http://openbookproject.net/thinkcs/python/english3e/fruitful_functions.html) *Fruitful functions*
* [Chapter 7](http://openbookproject.net/thinkcs/python/english3e/iteration.html) *Iteration*
* [Chapter 8](http://openbookproject.net/thinkcs/python/english3e/strings.html) *Strings*
* [Chapter 9](http://openbookproject.net/thinkcs/python/english3e/tuples.html) *Tuples*
* [Chapter 11](http://openbookproject.net/thinkcs/python/english3e/lists.html) *Lists*
* [Chapter 12](http://openbookproject.net/thinkcs/python/english3e/modules.html) *Modules*
* [Chapter 13](http://openbookproject.net/thinkcs/python/english3e/files.html) *Files*
* [Chapter 14](http://openbookproject.net/thinkcs/python/english3e/list_algorithms.html) *List Algorithms*
* [Chapter 15](http://openbookproject.net/thinkcs/python/english3e/classes_and_objects_I.html) *Classes and Objects - the Basics*
* [Chapter 16](http://openbookproject.net/thinkcs/python/english3e/classes_and_objects_II.html) *Classes and Objects - Digging a little deeper*
* *Dictionaries*

And I plan to continue covering the following topics before the end of class:

* [Chapter 10](http://openbookproject.net/thinkcs/python/english3e/events.html) *Event handling*
* *The Tkinter module* (for doing graphical interface programs without using Turtles)

It’s time to put all this great knowledge to use in a final project!

## The Requirements

1. **Code reusability is good**! This project gives you to opportunity to choose something from the course you really enjoyed and want to expand. In particular, you must enhance a program you or your partner has made in the course, and/or build off of code which we have provided to you.
2. Feel empowered to be creative! Your program or game should be interactive and as well as interesting to both create and to run/play. You can interact with the user via the interpreter, via the Turtle library, or through the Tkinter module to make the interactivity with the user interesting and fun.
3. Before you begin your program, you should create a Design Document detailing your plan. This must briefly describe which program or programs you modified, and why you chose this to do. Your Design Document is a plan; plans can change. But you should have a Design Document which outlines enough of your program that you feel confident you can complete all the work outlined by the deadline. **Because this is for planning before you begin coding, your Design Document is due earlier than the rest of the project, on April 10th.**

|  |
| --- |
| **Example: Software Design Document for Final Project****Our Proposed Project Extends the A12: DNA Assignment** by Dr. Scott Heggen and Dr. Jan Pearce  **MOTIVATION:** This program was motivated by an interest in the relationship between biology and computer science, and how computer science and algorithms can make biology problems easier, and in many cases, now solvable.  **PURPOSE**: The primary purpose of this software is to take a string, determine whether or not it is a valid DNA sequence, and if it is, return the corresponding amino acid sequence.  **RESOURCES**: This program is based on the A12: DNA assignment. We will be extending this assignment by adding in the following functionality.  **INITIAL DESIGN PLAN:**  We will complete this assignment by adding the following functions:  The original implementation of A12 included the following functions:   * is\_nucleotide() * num\_times() * complement\_strand() * mRNA() * chunk\_amino\_acid() * amino\_acid\_chunks() * genomics\_test\_suit() * testit() * main()   Our implementation for the final project will include:   * Refactoring the code to use classes. The CRC card for the class structure is included. * Implementing a new function called uncouple\_strand() which uncouples the DNA binding and strand scission activities of Fok I endonuclease. (Based on: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC47616/>)   **CRC Card**: see this link for a sample CRC card and a template to use for your own cards: [https://docs.google.com/document/d/1JE\_3Qmytk\_JGztRqkPXWACJwciPH61VCx3idIlBCVFY/edit?usp=sharing](https://docs.google.com/document/d/1ntNNd9f-1IpU1YQPVqLubWvubO5SroASfqDR6eQ6pU4/edit?usp=sharing)   * Because we are requiring obj ect-oriented design in your final project (see the next bullet), you will need to include a CRC card for your design. Again, this is a planning document, so a perfectly accurate CRC card is not expected. Plans can change! |

1. Your program must use object-oriented programming concepts in some way. So, your program may optionally be separated into as many modules as necessary. If you need multiple files, you must make sure to submit ALL the files needed to run your program or play your game, including any text files used by your program. Your design document above should include a CRC card explaining your planned object-oriented design.
2. Create instructions explaining how to use your program and/or play your game. You must at the very least include instructions as part of your accompanying report, but you can also integrate instructions into your program.
3. Include the usual descriptive header as a comment at the top of all of your Python files.
4. Final projects are designed to require more synthesis of the programming concepts than typical assignments. For this reason, they are more complicated, so are given more credit. For this project, create a two-page (or so) write-up that includes the following sections:
   1. **MOTIVATION**: In a paragraph, explain your motivation for choosing this particular project. *Include this portion in your design document due on April 10th.*
   2. **PURPOSE**: Briefly describe the primary purpose of your program. *Include this portion in your design document due on April 10th.*
   3. **INITIAL DESIGN PLAN**: This is the Design Document you created above. It should be created **before** your final write-up, but will also be included inside of the write-up in this section. It may not fully reflect your final product, as projects evolve over time. *Include this portion in your design document due on April 10th.*
   4. **SUMMARY**: A brief summary description of the design and implementation, including how much your initial design plan evolved, the final result you achieved and the amount of time you spent as a programmer in accomplishing these results, including any challenges overcome and innovations that were not specifically required by the assignment.. This should be no more than three paragraphs. *Include this portion in your final write-up due on April 24th*.
   5. **FILES**: A list in bullet form of the names of all files submitted (source code and input files, etc.). *Include this portion in your final write-up due on April 24th.*
   6. **VIDEO**: A publicly available link to a YouTube video which you create, demonstrating how to use your program. See below for the details on this section. *Include this portion in your final write-up due on April 24th*.
   7. **INSTRUCTIONS**: Explain how to use your program and/or play your game. *Include this portion in your final write-up due on April 24th*.
   8. **ERRORS**: A list in bullet form of all known errors and deficiencies that remain in your code. *Include this portion in your final write-up due on April 24th*.
   9. **REFLECTION**: A paragraph or two of your reactions to the final project. *Include this portion in your final write-up due on April 24th.*
   10. **REFERENCES**: Reference all ideas or code which are not your own, and describe and how you integrated the ideas or code into your program. *Include this portion in your final write-up due on April 24th.*

**NOTE**: Students often forget about the write-up, and try to do it at the very last minute. The result is usually lots of grammatical errors, which do count against you. Instead, I would encourage you to take breaks from coding by writing small sections of this document as you go. That will both help you take breaks from coding (which are useful in preventing exhaustion from trying to solve those annoying bugs) as well as ensure your document is in good shape by the end of the project.

1. **TESTING**: Your project should include a separate module containing a test suite, similar to what was explored in [T12: Modules](https://trello.com/c/GtLg6zi6). As a reminder, a good programmer follows Test Driven Development, and creates the test suite first!
2. A short (less than 5 minutes) demonstration video will also be required. This video will be presented during the last week of class. Your video should include:
   1. A title screen, including the class name, your name, and the project title
   2. You briefly discussing your project and it's motivation
   3. A demonstration on how to use the program, much like a marketing video.
   4. A short discussion about what you learned through the process of building the program.
   5. DO NOT include a section where you scroll through the code. Your video should include NO code.
   6. A credits roll
3. To submit your video, post the video to YouTube, mark it as public, and provide a link to the video in your final write-up.
4. Regardless of what you decide to do, you MUST include references to the original program and explain exactly how you modified and/or enhanced it in the report, as explained below. As always, ideas from the internet and all consultations with your peers, the TAs or instructors are encouraged, but MUST be documented.
5. While there seems like a lot of work around this final project, one thing to keep in mind is HAVE FUN with it! You are creating code to express your interests. Make a fun video. Break up the writing so it’s not so burdensome.

## Due Dates

* 1. Final Project Design Document due on April 10th, 2017 at 11:55PM.
  2. The rest of the Final Project (code, final write-up, and video) will be due by April 24th at 11:55PM.
  3. We will have a live demo session on our last day of class. It'll be the most fun final exam you've ever had!
  4. A point worth repeating: have a good time with your project and enjoy some creative freedom!

## Submission Instructions

1. Review the requirements above to ensure you have completed everything that was required of you. There is a lot; read it multiple times!
2. Save your code as **P0\_final\_*username*.py**. Replace *username* with your Berea usernames. For example, the TA Bianca Marrero’s file would be **P0\_final\_marrerob.py.**
3. Save your final write-up as **P0\_final\_username.pdf.** Replace *username* with your Berea usernames. For example, the TA Bianca Marrero’s file would be **P0\_final\_marrerob.pdf.**
4. Zip the code and the document together.
5. Upload the Zip file to Moodle by the due date listed on the course website: <https://trello.com/b/w7bIrLoV/>.
6. If you worked with a partner, your partner should upload a file named **P0\_final\_*usernames*.txt** and include both partner’s name in the document.