# BIOL 3312.001 Introduction to Programming for Biological Sciences

**Spring 2026**
**Classroom:** SCI 2.240 **Time:** MW 11:30am-12:45pm


| | | |
|---|---|---|
| **Instructor** | Dr. Faruck Morcos | Office BSB 12.601 |
| | **email**: faruckm@utdallas.edu | Office hours: by appointment |
| | Department of Biological Sciences | TA: TBD |
| | University of Texas at Dallas | **TA's email**: TBD@utdallas.edu |


## Pre-requisites

BIOL 2281: Introductory Biology Laboratory, BIOL 2311: Introduction to Modern Biology I and BIOL 2312: Introduction to Modern Biology II.

## Course Description

This course is an introduction to programming practices using Python designed specifically for students in the biological sciences. Special emphasis will be put in particular features of Python like object oriented programming, some data structures as well as applications to process, model and analyze biological data. The course will cover fundamentals of Biophyton, a popular set of tools for biological computation as well as the use of Python Notebooks as an environment to perform computations and run programs. One goal of this course is to provide a strong background on programming skills on a basic level while leaving more advanced techniques of software development and algorithms for other advanced courses.

## Outcomes

Students will be able to:

1. Be proficient with programming platforms and environments like Unix/Linux, Anaconda and Jupyter Notebooks which will allow an easier transition to other languages like AWK or Matlab.
2. Identify and use the most important structures needed to develop basic programs. These structures include data types, iterations and control structures.
3. Write programs and algorithms that analyze biological data like nucleotide and amino acid sequence data, expression data and molecular interaction datasets.
4. Build programs to model biological processes using the principles of object oriented programming.
5. Write efficient scripts that can be used to analyze and visualize the large amounts of genomic and molecular data deposited in local and public databases.

## Textbook and references

- Sebastian Bassi. *Python for Bioinformatics. 2nd Edition, 2017*
- Martin Jones. Python for Biologists. 2015 (**reference**)
- Neil Jones & Pavel Pevzner. An introduction to bioinformatics algorithms. MIT Press. 2004. (**reference**)

# Class Schedule

| | |
|---|---|
| *Jan. 21: Wed* | Computation and Life Sciences |
| *Jan. 26: Mon* | Introduction to Python |
| *Jan. 28: Wed* | Linux and data types |
| *Feb. 2: Mon* | Flow control |
| *Feb. 4: Wed* | Loops & Iterations |
| *Feb. 9: Mon* | Biological examples: interactive lecture |
| *Feb. 11: Wed* | Interactive lecture (cont.) |
| *Feb. 16: Mon* | Functions |
| *Feb. 18: Wed* | Function Parameters |
| *Feb. 23: Mon* | Modules and Packages |
| *Feb. 25: Wed* | Modules exercise |
| *Mar. 2: Mon* | Recursion |
| *Mar. 4: Wed* | Midterm |
| *Mar. 9: Mon* | Collections and Dictionaries |
| *Mar. 11: Wed* | Midterm Review |
| *Mar. 16: Mon* | *Spring Break (no class)* |
| *Mar. 18: Wed* | *Spring Break (no class)* |
| *Mar. 23: Mon* | File I/O and Sequence Conservation |
| *Mar. 25: Wed* | Sequence Conservation Interactive Lecture |
| *Mar. 30: Mon* | Sequence Conservation (cont.) |
| *Apr. 1: Wed* | Analysis of Sequence Conservation code and Logos |
| *Apr. 6: Mon* | Displaying Logos, packages and discussion |
| *Apr. 8: Wed* | Introduction to Biopython |
| *Apr. 13: Mon* | Common Data structures in Biopython |
| *Apr. 15: Wed* | Biophython In-class exercise |
| *Apr. 20: Mon* | Biophython In-class exercise (cont.) |
| *Apr. 22: Wed* | Introduction to Classes and Object Oriented Programming |
| *Apr. 27: Mon* | Interactive Lecture: Classes |
| *Apr. 29: Wed* | Interactive Lecture: Classes (cont.) |
| *May. 4: Mon* | Inheritance in OOP |
| *May. 6: Wed* | The Biological Molecule Class and Review |
| *Final Exam* | **Date:** TBD, **Time:** TBD , **Place:** TBD |

# Course Policies

## Grading

The grade is composed by a weighted average of the grades in midterm exam (15 %), in-class participation (5 %), a final exam (30 %), homework assignments (50 %).

## Homeworks

Programming homeworks will be assigned during the lecture. Homework guidelines will be posted on eLearning in a separate hand out.

## Collaboration Policy

Collaboration during homeworks and study sessions is highly recommended. Exchange of ideas is part of the learning process and programming is not an exception. However, **there should be a clear line between a healthy collaboration and unethical practices**. Homework problems can be discussed, debugging strategies can be shared as well as general ideas that could improve understanding of a topic or homework. **Writing code must be an individual task** and every submission should be easily reproducible by its author. **Copying code**, from another student or the internet, or **using AI tools** like ChatGTP, Gemini, CoPilot, etc. **is against the collaboration policy.** *Changing the name of variables* is **not** writing your own code :-). Groups of 2 students can be formed to officially collaborate in assignments, you must put the name of the person in your team in every submission.

## Exams

There will be **two exams**, a **midterm** exam and a **final** exam. Exams will have a combination of theory questions and programming problems. Given the nature of the material discussed in class, the final exam will be accumulative and will cover all the topics taught during the semester with an emphasis on the second part of the course.

## Attendance and Participation

Lecture **attendance** is an important part of your learning experience. Attending a lecture should be seen as a resource rather than an obligation. Attendance will not be strictly enforced but rather encouraged to improve the exchange of ideas, solve questions and clarify concepts. **Participation**, on the other hand, enriches the class experience and interactivity greatly improves learning connections. Participation also sets the pace of the lecture, a n important feature to avoid falling asleep! Attendance, participation as well as in class exercises or quizzes contribute to the final grade.

## Academic Honesty

Students are expected to follow the rules and guidelines of academic honesty established by the University of Texas. Basically, be honest about your contributions to homeworks, work on your own during exams and spend enough time reading and trying to understand class materials. This creed was voted on by the UT Dallas student body in 2014. It is a standard that Comets choose to live by and encourage others to do the same:

"As a Comet, I pledge honesty, integrity, and service in all that I do"

**Note:** The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.