



# Programming Abstractions in Python

---

CSE30 UCSC

INSTRUCTOR: DR. LARISSA MUNISHKINA

LECTURE: INTRO TO CLASS

# Agenda

---

- 1.Syllabus
- 2.Class Policies
- 3.Assignments (Notebooks) and Quizzes
- 4.Discussion Sections, Office Hours, Ed Forum
- 5.What you should know
- 6.What you will learn
- 7.Class Resources

---

# Part 1. Syllabus

---



- 
- All class material will be posted on Canvas <https://canvas.ucsc.edu>
  - **Canvas is the class primary resource:** you have to use it everyday
    - to check notifications
    - to access class material
      - lecture slides and movies
      - class webpages
      - lab assignments and notebooks
      - quizzes
      - surveys
    - to download and submit assignments
    - to participate in forums
    - to keep track of your grades!!!

# Syllabus

---

## 1. Part 1

1. Data Types
2. NumPy and Matplotlib
3. Data Collections

## 2. Part 2

1. Classes
2. Generators
3. Recursion

## 3. Part 3

1. Stacks and Queues
2. Trees
3. Graphs

# Class Policies and Grading

---

There are:

- 5 notebook assignments (NB)
- 10 quizzes

Extra Credit:

- TA/tutor sections (2 points each)
- extra credit labs/projects
- surveys

<b>Notebook Assignments</b>	<b>50%</b>
---------------------------------	------------

<b>Quizzes</b>	<b>50%</b>
----------------	------------

<b>Discussions and Extra Credit Labs</b>	<b>20%</b>
--	------------

# The Class Teaching Group

**Instructor:**

Larissa Munishkina (mlarissa@ucsc.edu)

**Office Hours:**

WF at 3:00 pm - 4:00 pm

<https://ucsc.zoom.us/j/3928413932>

**Lectures:**

MWF 9:20AM-10:05PM Humanities Lecture Hall

**Teaching Assistants:**

Alexei [apelyush@ucsc.edu](mailto:apelyush@ucsc.edu)

Jesh [jbheeman@ucsc.edu](mailto:jbheeman@ucsc.edu)

Nayan [nbhatia3@ucsc.edu](mailto:nbhatia3@ucsc.edu)

**Tutors:**

Leo [bli312@ucsc.edu](mailto:bli312@ucsc.edu)

Aryan [akhande1@ucsc.edu](mailto:akhande1@ucsc.edu)

Sruthi [slaravin@ucsc.edu](mailto:slaravin@ucsc.edu)

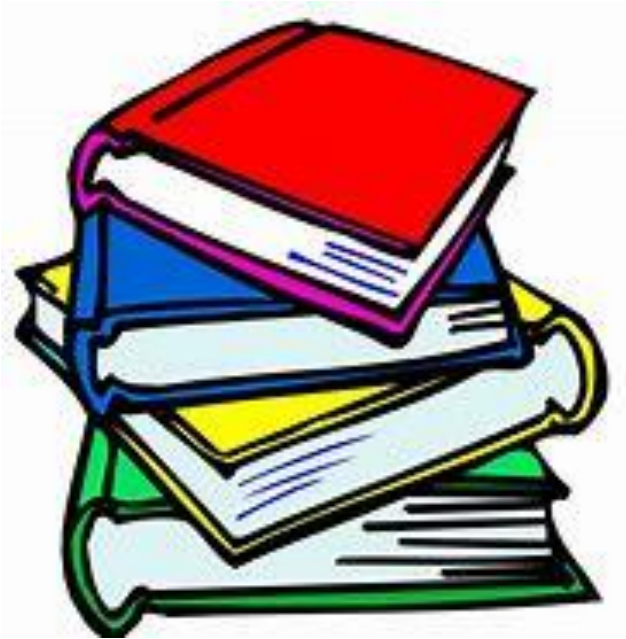
Chloe [cwong771@ucsc.edu](mailto:cwong771@ucsc.edu)

Waylon [wowillia@ucsc.edu](mailto:wowillia@ucsc.edu)

Daniel [dhong12@ucsc.edu](mailto:dhong12@ucsc.edu)

# Textbooks

---



## **Main Textbooks:**

[Programming Abstractions in Python |  
abstractions-in-python](#)

[CSE 30 Google Colab](#)

## **Optional Textbooks:**

[Problem Solving with Algorithms and  
Data Structures using Python](#)

[Python Cookbook, 3rd Edition](#)

[Introducing Python, 2nd Edition](#)

[Fluent Python, 2nd Edition](#)

[Learning the Unix Operating System](#)



# Additional Resources

---



[The Python Standard Library — Python 3.8.9 documentation](#)

[NumPy](#)

[SciPy](#)

[TkDocs Tutorial](#)

[Gallery — Matplotlib 3.4.1 documentation](#)

[turtle — Turtle graphics — Python 3.8.9 documentation](#)

[Pygame Front Page — pygame v2.0.1.dev1 documentation](#)

[UNIX / Linux Tutorial for Beginners \(surrey.ac.uk\)](#)

---

# Part 2. Getting Help

---

ED AND NOTEBOOK GRADER





# Getting Help!!!

---

- **Office Hours:**
  - WF at 3:00 pm - 4:00 pm
  - <https://ucsc.zoom.us/j/3928413932>
  - Meetings will be one-on-one
  - You should wait in the waiting room to be admitted
- **Discussion Sections – Secondary Labs**
  - Are posted on Canvas
  - You should attend 1 section a week
- **Canvas Posts**
- **Ed Forum**



# TA Sections

## Associated Discussion Sections or Labs

#11322 LBS 01A Enrl: 40 / 40	M 11:45AM-12:45PM Wait: 1 / 999	Staff ▲ Wait List	Loc: J Baskin Engr 105
#11321 LBS 01B Enrl: 36 / 40	M 01:00PM-02:00PM Wait: 1 / 999	Staff ● Open	Loc: J Baskin Engr 105
#11325 LBS 01C Enrl: 39 / 40	Tu 02:30PM-03:30PM Wait: 0 / 999	Staff ● Open	Loc: J Baskin Engr 105
#11324 LBS 01D Enrl: 38 / 40	W 01:15PM-02:15PM Wait: 0 / 999	Staff ● Open	Loc: J Baskin Engr 105
#11323 LBS 01E Enrl: 32 / 40	Th 09:45AM-10:45AM Wait: 0 / 999	Staff ● Open	Loc: J Baskin Engr 105
#11320 LBS 01F Enrl: 39 / 40	Th 11:00AM-12:00PM Wait: 0 / 999	Staff ● Open	Loc: J Baskin Engr 105
#26024 LBS 01G Enrl: 1 / 40	W 02:15PM-03:15PM Wait: 0 / 999	Staff ● Open	Loc: J Baskin Engr 105



# Ed Discussion Forum

---

We will be using Ed <https://edstem.org/us/courses> as a question-answering forum.

If you have questions concerning your homework, programming assignments, exams, or lecture material or want to share your ideas or information, post it on Ed.

The instructor, teaching assistants (TAs) and tutors will answer them ASAP.

If you know the answer to other student's question you can also post your answer.

**Ed is a place to get help, help others, or discuss ideas, projects, etc.**





# Posting Code on Ed

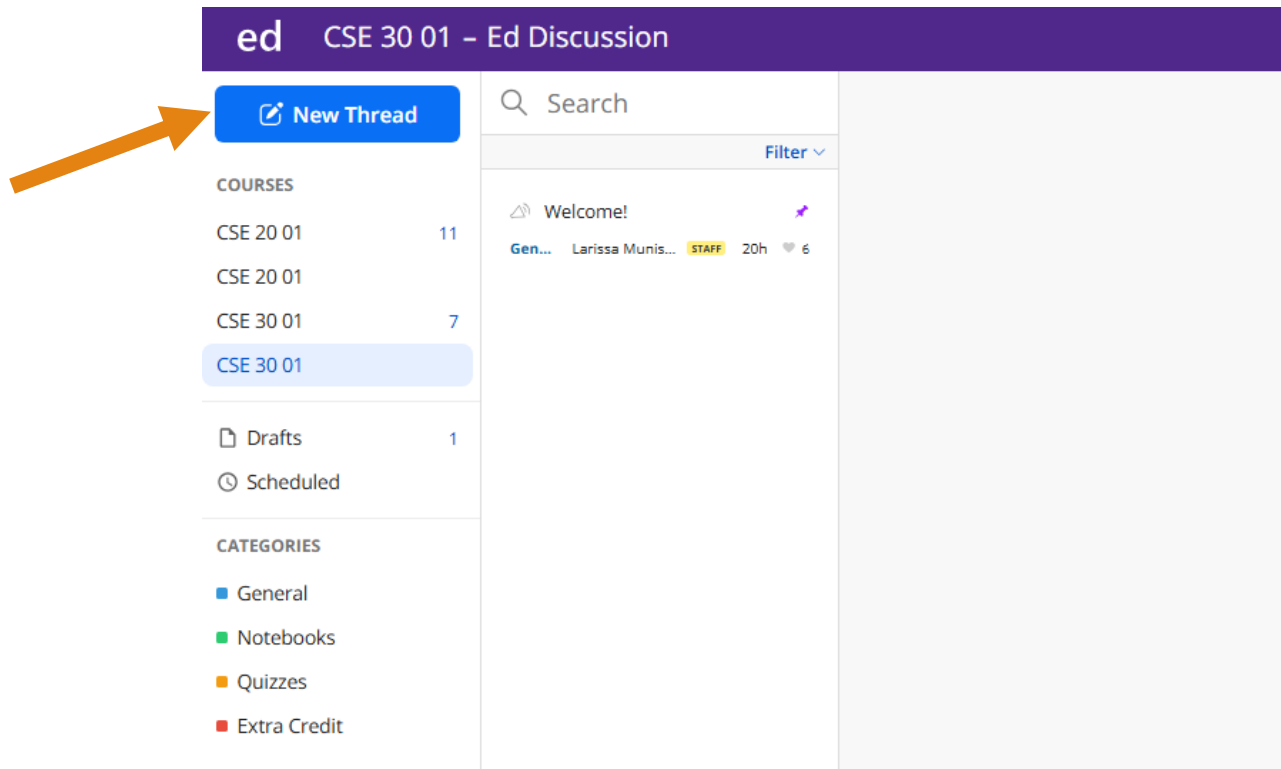
1. On Canvas, select Ed Discussion.

A screenshot of the Canvas LMS interface for a course titled 'CSE-30-01'. On the left is a vertical navigation menu with items: Home, Syllabus, Grades, Assignments, Quizzes, Files, Ed Discussion (highlighted with a blue box and an orange arrow pointing to it), YuJa, Announcements, Discussions, People, and Modules. The main content area is titled 'Recent Announcements' and lists three announcements: 'CSE 30 Tutoring Hours and Course Discord', 'Zoom section for 1/6 Monday', and 'Welcome to CSE 30 class!'. Each announcement includes a user profile picture, a 'Reply' link, and a 'Posted on' timestamp. On the right side, there is a sidebar with buttons for 'Import Existing Content', 'Import from Commons', 'Choose Home Page', 'View Course Stream', 'Course Setup Checklist', 'New Announcement', 'New Analytics', and 'View Course Notifications'. At the bottom right, there is a 'Coming Up' section with a 'View Calendar' link and the text 'Nothing for the next week'. At the top right of the course page, there are buttons for 'View as Student' and 'Immersive Reader'.



# Posting Code on Ed

2. On Ed Discussion (Ed), select New Thread.





The screenshot shows the Canvas LMS interface for creating a new post. The top navigation bar includes the Canvas logo and course information. The left sidebar contains a 'New Thread' button and lists of courses and categories. The central area shows a 'Welcome!' message. The right panel, titled 'New Post', allows users to create different types of content (Question, Post, Announcement). The 'Post' tab is selected, and the title 'Help me with NB 1' is entered. The category 'General' is selected from a row of options. Below the categories is a rich text editor toolbar with various icons for formatting and inserting content. An orange arrow points to the code icon in the toolbar.





# Posting Code on Ed

4. Write (type or paste) your code in the code window.
5. You can run the code (notice that the code generates an error, do you know why ?)

The screenshot shows the 'New Post' form in the Ed application. At the top, there are buttons for 'Cancel', 'New Post', 'Schedule', and 'Post'. Below these is a 'Title' field containing 'Help me with NB 1'. Under the 'Category' section, there are four buttons: 'General' (selected), 'Notebooks', 'Quizzes', and 'Extra Credit'. The main content area features a rich text editor with a toolbar containing icons for paragraph, bold, italic, underline, code, link, unlink, list, indent, image, link, unlink, link, unlink, link, unlink, and a menu icon. Below the editor is a code window. The code window has a 'Run' button (a blue button with a play icon) which is pointed to by an orange arrow. To the right of the 'Run' button are checkboxes for 'Line Numbers' and 'Runnable', both of which are checked. To the right of these checkboxes is a language dropdown menu set to 'Python' and a full-screen icon. The code window contains the text: 

```
1 print('Hello!' + 5)
```

# Notebooks and Notebook Grader Website

---

All required assignments should be done on the notebook grader website:




<https://notebookgrader.com/>

The assignments are in the format of Jupiter notebooks (NB) with the file extension ipynb.

The assignments are automatically loaded into your Google drive, so you can access them directly through the Google drive.

The assignments are autograded on the notebook grader website.

You need to submit your notebook, so your grade is recorded on the website (Click the button Grade My Assignment).

 NotebookGrader  Student  Teacher

## Data Structures

Submission deadline: ⓘ Sat, Dec 7, 2024, 11:59 PM  
Late submission possible until: ⓘ Sat, Dec 7, 2024, 11:59 PM

### Your Assignment

Access assignment

If you corrupt your copy of the assignment, you can [get a new copy](#).

### Grades ⓘ

You can have your work graded 4 more times in 24h. Save your Colab notebook before requesting grading.

Grade My Assignment

# Data Structures

Submission deadline: ⓘ Sat, Dec 7, 2024, 11:59 PM

Late submission possible until: ⓘ Sat, Dec 7, 2024, 11:59 PM

## Your Assignment

Access assignment

If you corrupt your copy of the assignment, you can [get a new copy](#).

You can get the feedback (for debugging) by clicking on the Review button.

## Grades ⓘ

You can have your work graded 3 more times in 24h. Save your Colab notebook before requesting grading.

Grade My Assignment

Valid	Grade	Graded On	Graded Assignment
✓	0.00 / 100.00	Sun, Sep 29, 2024, 8:52 AM	<div>Review</div>



---

# Part 3. What You Will Learn

---

LABS AND PROJECTS

PROGRAMMING ENVIRONMENT

CODING ANTIPATTERNS

A solid orange horizontal bar at the bottom of the slide.

# What You Should Know

---

## Data Types

- Primitive Data Types
  - None
  - Boolean
  - Integer
  - Float
  - Complex
  - String
- Data Collections
  - Lists
  - Arrays
  - Dictionaries
  - Tuples
  - Sets
  - Frozensets

## Control Flow

- Conditional Statements
  - if
  - if-else
  - elif
  - nested conditional statements
- Loops
  - for
  - while
  - nested loops
- Functions

## OOP

- Classes
- Encapsulation
- Inheritance
- Polymorphism

# What You Will Know

## Abstract Data Types

- Stack
- Queue
- Deque
- Priority Queue
- Hash Table
- Heap
- Tree
- Graph

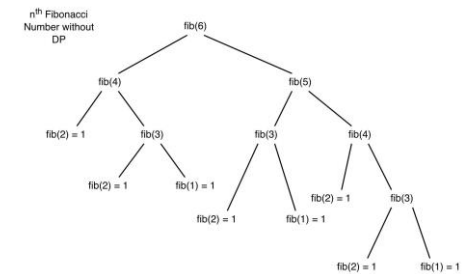
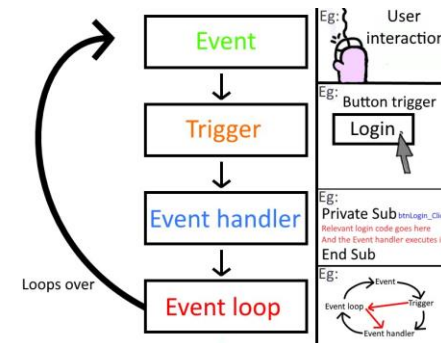
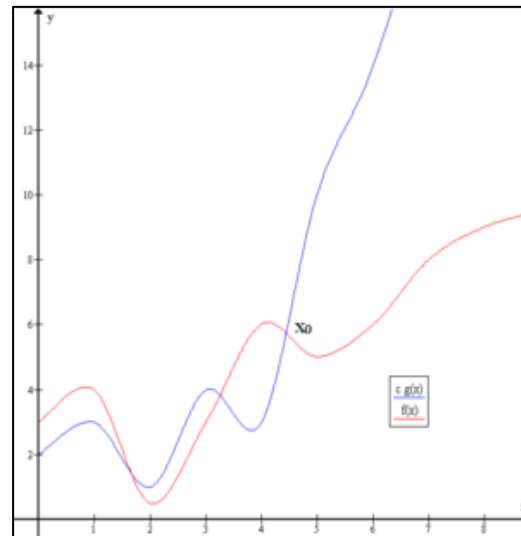
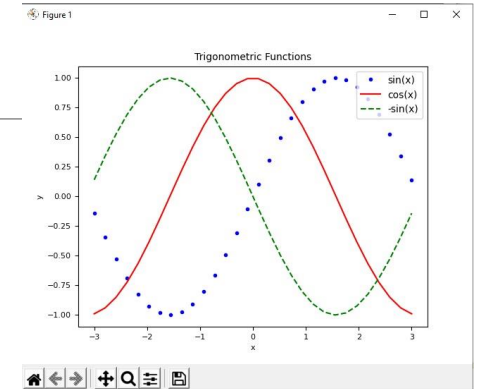
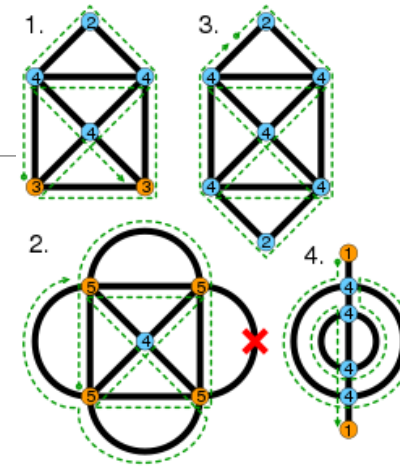
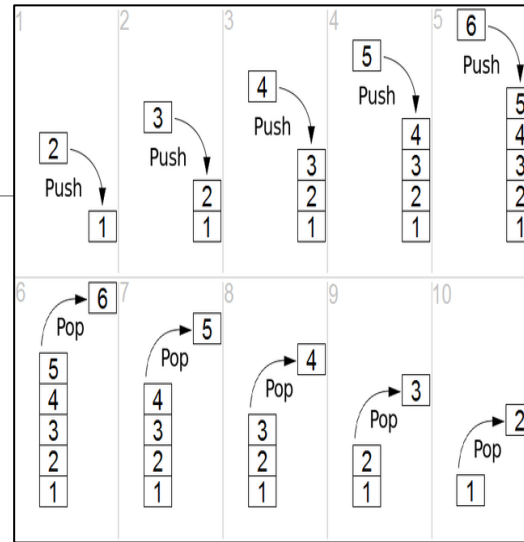
## Control

- Generators
- Recursion
- (Optional) Functional Programming
- (Optional) Event-Driven Programming
- (Optional) Threading

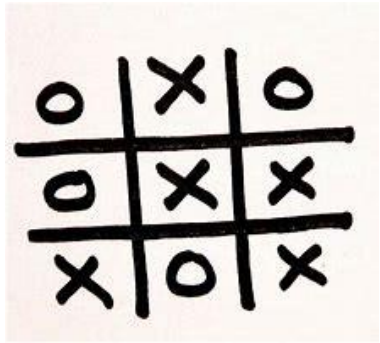
## (Optional) Algorithms

- Big O Notation
- Sorting
- Search
- Optimization
- Divide-and-Conquer
- Greedy
- Flow Network
- Dynamic Programming
- Linear Programming

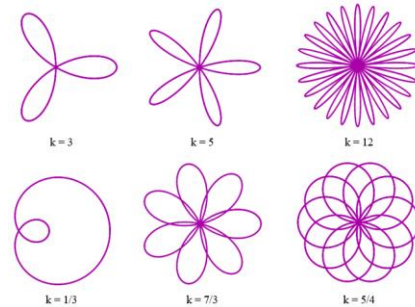
ADTs  
Graphs  
Plotting  
Big O  
Events  
Recursion



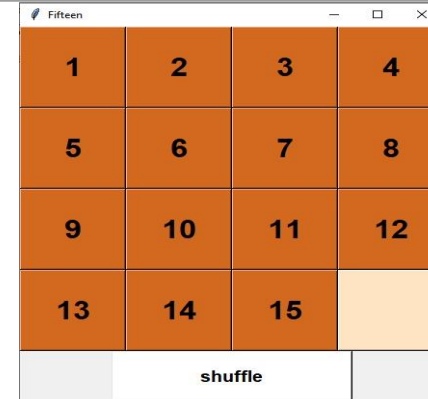
# Projects and Labs



Tic Tac Toe



Rose Curve



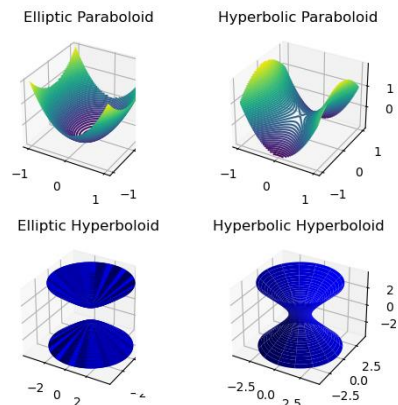
Fifteen Puzzle



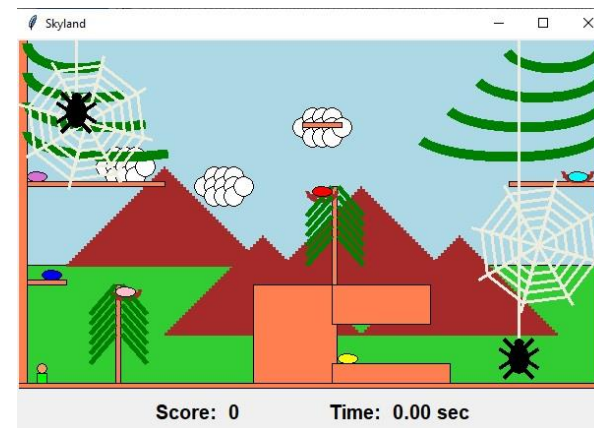
n-Queens Problem



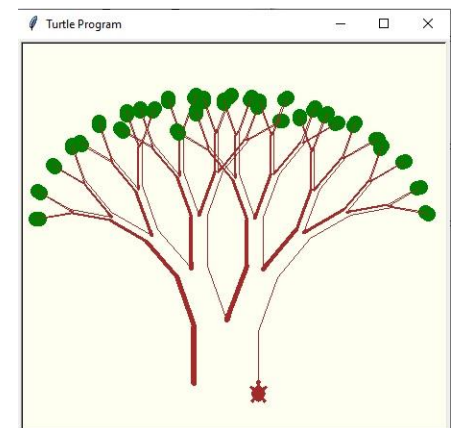
Calculator



Quadric Surfaces



Video Game



Tree Fractal



# What Platform to Choose?



- zyBooks, Codio, coding online

- Professional IDE



- Notebooks or Colab



- IDLE

- What programming platform to choose is a difficult question.
- As a beginner, you may need to ride a tricycle – you may learn a simple IDE such as IDLE.
- If you like the virtual reality, you may prefer coding online.

# Programming Platforms

---

## **Simple IDE such as IDLE, Notepad ++, TextPad**

- easy to handle and learn, no help on managing projects

## **Professional IDE such as PyCharm, VSCode, Eclipse, Atom, Sublime, Vim, Emacs**

- difficult to learn, can help to manage projects

## **Interactive shells such as iPython, Jupyter Notebook, or Colab**

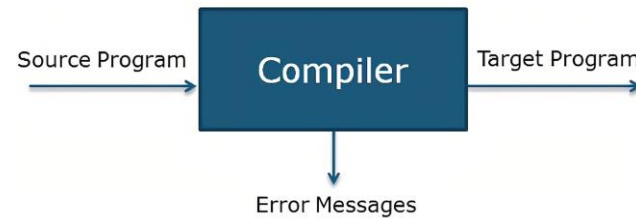
- nice interfaces, easy or difficult to learn depending on your task, the only problem is that it is a shell, not an IDE

## **Online coding platforms such as zyBooks, Codio, or Codecademy**

- nice interfaces, easy or difficult to learn depending on your task, the only problem is that it is not an IDE

# Programming Process

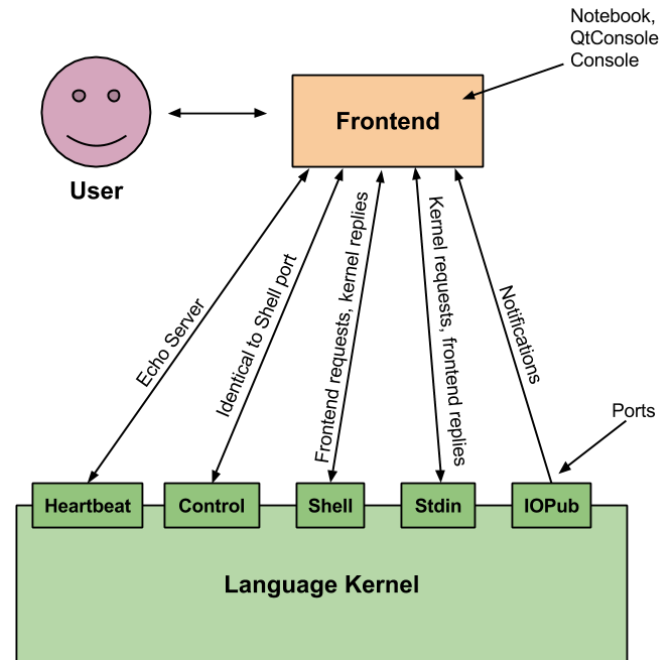
```
1 #include<stdio.h> //standard input output
2 #include<conio.h>
3
4 main()
5 {
6     int price;
7     float tax,final;
8     printf("Enter the price:\n");
9     scanf("%d",&price);
10    tax=price*price*0.15;
11    final=tax-tax*0.1;
12    printf("%d\n",final);
13 }
```



Shells are used for interactive programming or command executions.

Compilers are used to create a standalone application that can run in the OS shell.

Jupyter, Colab, Codio have a webpage interface to provide interactions with the user – they are not designed for generating a standalone application.



# Programming vs Scripting

---

You can watch the video about programming languages vs scripting languages here: <https://youtu.be/g0Q-VWBX5Js> .

Programming usually means to write a program, whereas scripting means to write a script.

Scripts are usually smaller in size and based on libraries, whereas programs are long and divided into modules.

Scripts usually do not have sophisticated control flow structures (people try to avoid them), whereas program are based on control flow structures.

Scripts depend on the parental application (their kernel), whereas programs are standalone and written from scratch – you can write a program that is a shell for scripting, you cannot do it with scripts.

# Bad Coding Practices - Coding Antipatterns

---

You can read about coding antipatterns here:

- [8 Coding Antipatterns Every Web Developer Should Avoid in Web Development Process \(dngwebdeveloper.com\)](http://dngwebdeveloper.com)

## Spaghetti Code

- usually means disorganized or incorrectly organized code that is difficult to read and understand

## Hard Coding

- means embedding data into source code

## Golden Hammer

- means overusing a platform or approach for all tasks

## Boat Anchor

- means designing code that is not needed



# Homework & Assignments

---

Get familiar with resources on Canvas.

Set up your working environment (Check the notebook grader website).

Post messages on the Ed Forum.