

# Practice Problems

## CS 1301 - Intro to Computing - Fall 2024

### Important

---

- Resources:
  - TA Helpdesk
  - Email TA's or use Ed Discussion
  - Textbook: [How to Think Like a Computer Scientist](#)
  - [CS 1301 YouTube Channel](#)
  - Handouts (in Canvas Files)
- Comment out or delete all function calls. Only import statements, global variables, and comments are okay to be outside of your functions.
- **Read the entire document before starting this assignment.**

### Purpose

---

The purpose of the practice problems is to strengthen your coding skills and prepare you for the coding section of your exams. The problems are created to reflect the difficulty level of problems you might see on an exam. They will challenge you to combine and apply the different concepts you have learned in class. We will not be providing a skeleton file to assist you with completing the practice problems. Please name your file `practiceProblems.py` before submitting to Gradescope!

### Grading

---

You are given the opportunity to earn 0.5% of extra credit applied to your final grade by successfully completing all of the practice problems. Practice problems will be progressively released throughout the semester, but in the end, there will be a total of 20 practice problems for 100 auto-graded points. The amount of extra credit (as a percent) you will earn towards your final grade can be calculated by:

```
extraCreditPercentage = autogradedPoints / 200
```

Remember, not all points will be obtainable at this time. You must complete *all* releases of practice problems to earn full extra credit.

The deadline to submit practice problems for extra credit is **Tuesday, December 3<sup>rd</sup> at 11:59pm**.

# Week 1: Functions and Expressions

---

## Rock, Ramble, and Roll at the Student Center

**Function Name:** rockRambleRoll()

**Parameters:** N/A

**Returns:** None

**Description:** You have arrived at the Student Center for this year's Rock, Ramble, and Roll, and you are tasked with ordering food for your friends. At the Student Center, pizza at Campus Crust costs \$9.00, sandwiches at 5th Street Deli cost \$5.50 each, sushi rolls at Bento Sushi costs \$8.00 each, and tacos at Twisted Taco cost \$2.25 each. Write a function called `rockRambleRoll()` that asks the user how many pizzas, sandwiches, sushi rolls, and tacos you need to order for you and your friends. The function should then print the total cost of all of the food ordered.

**Note:** You can assume all inputs will be integers.

```
>>> rockRambleRoll()
How many pizzas do you want at Campus Crust? 5
How many sandwiches do you want at 5th Street Deli? 2
How many sushi rolls do you want at Bento Sushi? 1
How many tacos do you want at Twisted Taco? 7
You need to spend $79.75 at Rock, Ramble, and Roll.
```

```
>>> rockRambleRoll()
How many pizzas do you want at Campus Crust? 1
How many sandwiches do you want at 5th Street Deli? 0
How many sushi rolls do you want at Bento Sushi? 1
How many tacos do you want at Twisted Taco? 0
You need to spend $17.0 at Rock, Ramble, and Roll.
```

## House Party at Bobby Dodd Stadium

**Function Name:** houseParty()

**Parameters:** N/A

**Returns:** None

**Description:** You are excited to visit Bobby Dodd Stadium for their annual house party, but you need to make sure to leave enough time to visit the evening mixer at the exhibition hall. Each activity at the house party takes a set amount of time: each autograph takes 10 minutes, each visit to the bouncy castle takes 20 minutes, each video game contest takes 45 minutes, and walking through Georgia Tech's locker room takes 15 minutes. Write a function called `houseParty()` which asks the user how many times the activities listed above will be done. This function should then print out the total time in hours and minutes the user will spend at the house party.

**Note:** You can assume all inputs will be integers.

```
>>> houseParty()  
How many autographs will you get? 5  
How many times will you visit the bouncy castle? 1  
How many video game contests will you participate in? 0  
How many times will you walk through the locker rooms? 1  
You will spend 1 hour(s) and 25 minutes at the house party.
```

```
>>> houseParty()  
How many autographs will you get? 6  
How many times will you visit the bouncy castle? 3  
How many video game contests will you participate in? 0  
How many times will you walk through the locker rooms? 4  
You will spend 3 hour(s) and 0 minutes at the house party.
```

## Week 2: Conditionals

---

### Open Sesame!

**Function Name:** enterTheCave()

**Parameters:** phrase ( str ), code ( int )

**Returns:** None ( NoneType )

**Description:** You are part of a group of hackers who have a secret hideout in a cave. A few months ago, your passcode was leaked, and a civilian managed to break in. You have been asked to improve the security of the cave by adding another layer of complexity to the cave's entrance system. Write a function that takes in a phrase ( str ) and a code ( int ) . If the phrase is "Open Sesame" and the code is between 0 and 25 (both inclusive), then you open the door. This means the function should **print** "You have opened the door" . The door can also be opened with the phrase "Hello World" and the code between 75 and 100 (both inclusive). If the phrase is "Python Enjoyer" and the code is exactly 50, the function should **print** "You have closed the door" . If the phrase and code combination is outside of these possibilities, the function must **print** "INTRUDER ALERT" .

```
>>> enterTheCave("Open Sesame", 22)
You have opened the door
```

```
>>> enterTheCave("Hello World", 50)
INTRUDER ALERT
```

## Dinner Time

**Function Name:** `dinnerTime()`

**Parameters:** `xCoordinate ( int )`

**Returns:** `closestRestaurant ( str )`

**Description:** The sun is setting and it is time for you and your friends to grab dinner at the Bright Beach Boardwalk. Since you have already been out and about the entire day, your group decides to eat at the closest nearby restaurant. Write a function that takes in your group's current x-coordinate on Bright Beach Boardwalk. This function should return the restaurant that is the closest on the boardwalk to your group. Reference the table below for the locations of various restaurants on Bright Beach Boardwalk.

**Note:** You can assume there will be no ties.

Restaurant	X-Coordinate
Sun, Sand, and Seafood	250
The Beachcomber's Bistro	460
Coastal Catch	820
Beachside Bonanza	870
Tidal Wave Tavern	940

```
>>> dinnerTime(800)
'Coastal Catch'
```

```
>>> dinnerTime(310)
'Sun, Sand, and Seafood'
```

## Week 3: Iteration

---

### Upside Down Light Communicator

**Function Name:** lightCommunicator()

**Parameters:** upsideDownMessage ( str )

**Returns:** message ( str )

**Description:** To communicate with your friends in the Upside Down, you need to reverse the messages you are receiving as well as flip the S's and the T's (case insensitively). Given the message you have received, return the translated message.

```
>>> lightCommunicator("!EDIH - gnimoc ti nogrogomed ehS")
'The demogorgon is coming - HIDE!'
```

```
>>> lightCommunicator("!emoh emoc ,llIW")
'Will, come home!'
```

### Binge Time

**Function Name:** totalTime()

**Parameters:** episodeRuntimes ( str )

**Returns:** totalWatchTime( str )

**Description:** The new season of Ginny and Georgia just came out and you cannot wait to catch up on all the drama! Given a string of episode run times (each separated by a space), return the number of hours and minutes it will take you to binge all episodes as a string. The returned string should be of the form "{ } hour(s) and { } minute(s)" , and you may assume you will always have a whole number for the number of minutes.

```
>>> totalTime("87 100 70")
'4 hour(s) and 17 minute(s)'
```

```
>>> totalTime("50 66 90 43")
'4 hour(s) and 9 minute(s)'
```