CS 1112: Introduction To Programming

Exam 1 Review

Dr. Nada Basit

basit[*at*]Virginia[*dot*]edu

Friendly Reminders

- Your ***safety*** and ***comfort*** is important!
  - If you choose to wear a mask you are welcome to do so
  - *We will interpret wearing a mask as being considerate and caring of others in the classroom (not that you are sick), and realize that some may choose to mask to remain distanced*
- Be an ***active*** participant in your learning!
  You're welcome and ***encouraged*** to ask questions during class!
- If you feel ***unwell***, or think you are, please stay home
  - *We will work with you!*
  - Get some rest ☺
  - View the recorded lectures – *please allow 24-48 hours to post*
  - *Contact us!*

Announcements

- **Quiz 1, 2, 3, 4** transferred from Sherlock to Canvas
- **PA03 (Functions)** graded (on Gradescope and Canvas)

- **Exam 1** is on February 28, 2024 (Wednesday)
  - If you have **SDAC** *time and/or distraction-free accommodations*, please **book** a time slot with SDAC to take the exam at their facility (*any time* on the same day, not another day, please, unless you've spoken with me ahead of time!)

  - Everyone else: please come to class as normal! ☺ *(Please show up on-time!)*

Exam 1: Policies

- Closed-book, closed-notes,... closed everything ☺
- Closed PyCharm (IDE)
- No collaboration at all - *must represent your individual effort*
- Location: Taken in-class (in-person)
- Exam taken on Sherlock
- Duration: class time - 1 hours and 15 minutes
- Review session: Tuesday night
  - (Attendance is optional. Time & Location coming soon!)

Exam 1: Policies

- Closed-book, closed-notes,... closed everything ☺
- Closed PyCharm (IDE)
- No collaboration at all - *must represent your individual effort*
- Location: Taken in-class (in-person)
- Exam taken on Sherlock
- Duration: class time - 1 hours and 15 minutes
- Review session: Tuesday night
  - (Attendance is optional. Time & Location coming soon!)

Please bring your UVA ID card with you to the exam!
Check-in with TA as you leave!

Exam 1: Resources to Study From

- Quizzes 1, 2, 3, and 4
- Programming Assignments PA00 – PA03
  - Don't worry about PA01 (Turtles)
- Course Materials on Canvas (all: incl. Style & Debugging)
  - Except slides relating to Turtles
- Python Scripts on Canvas (all)
  - Except scripts relating to Turtles
- In-class activities (all)
  - … you guessed it…
    except activities relating to Turtles

No Turtles on Exam 1

Don't forget about me! ☹

General

- Emphasis on content and reading / understanding code
- Only asked to write a very small amount of code
- Read code and answer questions about it
- Given input, trace code, what is the output / what does the code do?
- Given code, what might be wrong with it?
- Provide an example scenario for a given programming concept

Basics

- `print()` function, concatenation (+)
- Print with end and sep
- Comments
  - ‘‘‘comment’’’
  - """comment"""
  - # comment
- Escape characters: \n \t

Basics

We have seen examples that uses **lists** as part of the code, like in a for-loop, etc. so in this context we may see lists.

- Variables
  - Different value types: int, float, Boolean, string, list (not: tuple, dictionary)
  - type(x): gives the data type
- Arithmetic operators: + - * / // ** %
  - //: integer division
  - **: power (a^b)
  - %: modulus
- Assignment statement
- Assignment vs asking about Equality (= vs ==)
- Boolean Operations: and or not

Basics

- Type Casting
  - E.g.,
```
f = 7.11
int_f = int(f)
```
  - Beware of invalid type casting
  - E.g.,
```
text = "42abc"
num = int(text)
```
- Pseudocode
  - Writing and reading/understanding pseudocode

The building blocks of programs [Day 3, slide 12]

- Sequence
- Repetition
- Conditions / decisions
- Named actions

Algorithms

- What makes a good algorithm? [Day 3, slide 19]
  - Unambiguous
  - Executable
  - Terminating
  - Deterministic
  - Correct
- What are bad algorithms?
  - Ambiguous
  - Not executable
  - Non-terminating
  - Incorrect

input() function

- Obtaining information from the user
- Input obtained from the keyboard
- Data transferred and stored as a string
  - Type case afterwards if needed
- E.g., `num_of_sales = input("How many items did you sell?")`

Type cast: `num_of_sales = int(input("How many items did you sell?"))`

Booleans and Conditionals

- Booleans
  - Boolean data type
  - Values (True and False)
  - Operators (not, and, or)
  - Comparison operators
    (< > <= >= == !=)
  - Boolean expression / relational expression
- Conditionals
  - Conditional decision statement
  - Examples:
    - if
    - if-else
    - if-elif
    - if-elif-else
    - match-case

Mutability

- Strings are immutable!
  - Cannot change the string (cannot modify it)
  - You can swap out a string for another
    (a new string can include pieces of old strings)
- Various types and their mutability [Day 11, slides 12-17]

Type	Stores	Syntax	Mutable?
Range	ints	`range(3,7)`	no
String	characters	`"Hello", "abc 123"`	no
List	anything	`[1,2,3,6,"hello"]`	yes
Tuple	anything	`(1,2,3,6,"hello")`	no

FUNCTIONS (1 of 3)

- Reasons to use functions [Day 12, slide 6]
- Function header / creating functions / parts of a function [Day 12, slides 7-8]
- Indentation (part of the syntax)
- `return` statement
- Calling functions - function call
- Parameters vs arguments
- Side effects of functions
- Use informative function names

FUNCTIONS (2 of 3)

- Comment your code (in-line), additionally use: multi-line docstrings
- You can save useful functions in a file for reuse in the future
- Write code clearly and concisely
  - Increase readability
  - Increase ease of debugging / troubleshooting
  - Increase ease of writing the code in the first place
- Functions calling other functions
- Scope
  - Global vs local variables ("car with tinted windows")
  - Recognize and identify local and global variables in given code

FUNCTIONS (3 of 3)

- Function arguments [Day 14 (most of these slides)]
- Arguments are
  - Positional
  - Named `my_func(a=10)`
- Parameters are
  - Required, like "a"
  - Optional, like "b", default value provided
  - `def another_func(a, b=17)`
- Mixing positional and keyword (named) arguments
  - Positional arguments (all) should come before any keyword arguments

Loops

- For-loops [Day 15]
  - `for <variable> in <collection>`
    - Strings
    - Lists
    - Integers (range)
  - Remember indentation
  - for-else
- While-loops
  - Guard condition (boolean expression)
  - Some aspect of the guard condition must change in the body of the loop
  - Remember indentation
  - While-else

- When to use each kind [Day 15, slide 48]
- Break
- Continue

Testing/Debugging & Style

- Debugging Techniques:
- `print()` statements
- Calling functions: expectation vs. actual
- Trace code
- Commenting should help in this endeavor
- Helpful to name variables and functions appropriately to increase readability (which in tern aids in debugging)
- Style - helps in readability and debugging

Q&A

I’m happy to address any questions you have about the exam!

Reminder: CS Laptop Loaner Program

- This course requires students to have a **laptop**
- I realize that not everybody might have one (nor necessarily need one for their desired major / path…)
- If you do not have a laptop for any reason… ***not to worry!***
- The CS department's Systems staff has a notebook / laptop loaner program and will be able to loan you a notebook / laptop computer for the duration of the semester if you don't have one or if you cannot afford one.
  - Also available if your laptop is broken and under repair, we can arrange for you to receive a loaner laptop for a week or two until your own laptop is fixed

Interested? Link: https://www.cs.virginia.edu/wiki/doku.php?id=cs_laptop_loaner

I am happy to be your sponsor. Please let me know.