Lecture 34:

CLOSING REMARKS

CSC111: Introduction to CS through Programming

R. Jordan Crouser

Assistant Professor of Computer Science

Smith College

- Administrivia: FP4 and FP5
- Semester in review
- “Half-hour hackathon”

FP4: Finished Version (i.e. code)

- **Complete, organized, well-documented code**:
  - in-line **comments**
  - appropriate **variable / function names**
  - **docstrings** explaining what each function / class does
  - includes all **supplemental files** (images, etc.)
- Include a text file named **README** containing:
  - a **description** of the project
  - information on **how to run** your project
  - install links to any **required modules**
  - a list of **known bugs**
- **Runs without syntax errors**

FP5: Write-up

- Due Monday December 14th at 11:55pm
- A short (2-3 page) paper, including:
  - a description of the **project**, including the **intended audience**
  - a description of the **architecture** (how you designed the code)
  - a discussion of at least **one major challenge**, and how you overcame it
  - a description of **your specific contributions** to the project
- Two options:
  - Each team member writes their own (most common)
  - One collaborative submission (also perfectly fine)

Some perspective…

it’s been **96 days** since we started programming

Some perspective…

we had **34** lectures,

10 labs, & **6** assignments

Some perspective…

together we wrote more than **127,311** lines of code

(**not** including your final projects)

Some perspective…

that’s roughly 1,286 lines

per person

(>20 printed, single-spaced pages)

Some perspective…

or **~3,700 lines** for every day that we met

We have covered a LOT of ground…

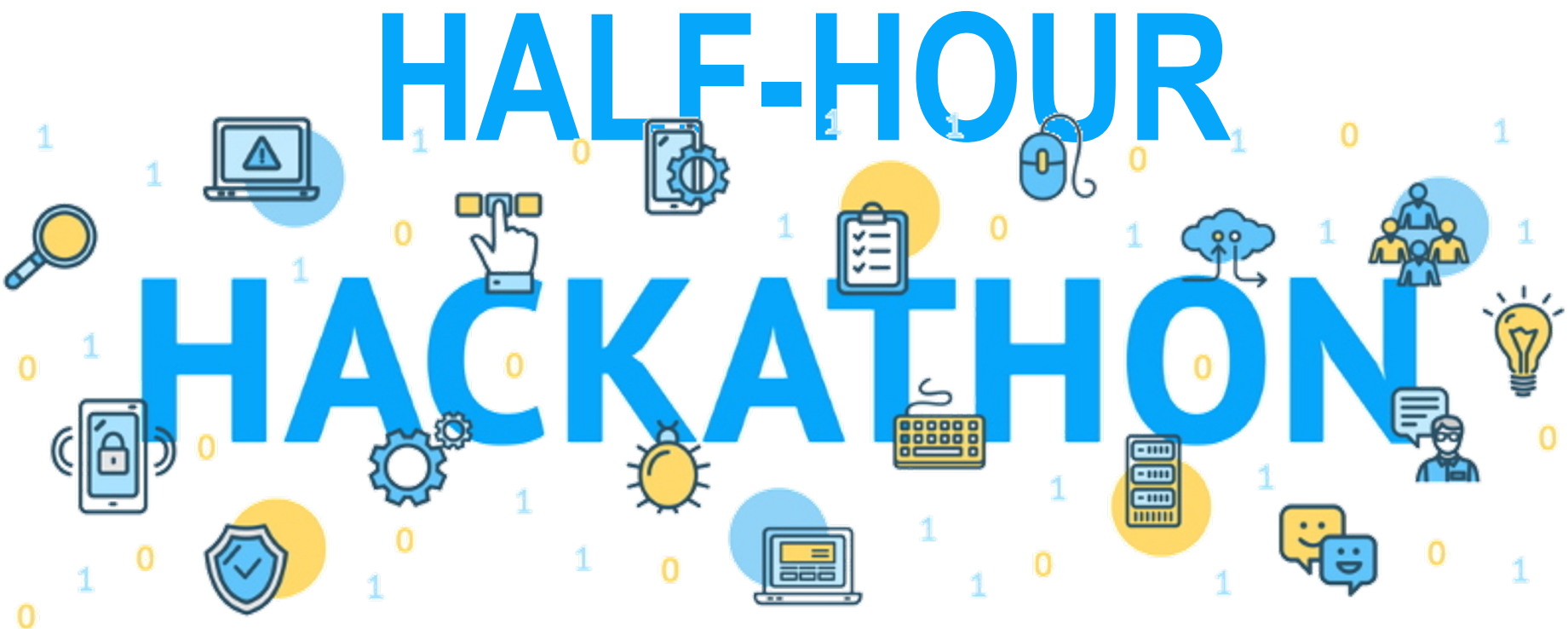
- Basic hardware (processor, RAM, hard drive, etc.)
- Data types (`int`, `float`, `string`, `bool`, `list`, `dict`) and the operations they support
- Useful built-ins: `print`, `range`, `eval`, `input`
- Exceptions and how to handle them
- Conditional statements
- Loops (`for` and `while`)
- Defining / calling functions
- Handy packages (`math`, `random`, etc.)

We have covered a LOT of ground…

- Data structures
- Recursion
- Object-oriented programming
  - Classes
  - Inheritance
- Algorithms and efficiency
- Working with files
- Graphics, animation, and interaction
- Ethical issues in CS

Discussion

What do you think you will remember about this course in **5 years**?


HALF-HOUR
HACKATHON

How it works

- **Objective**: write a short program to commemorate your group's **experience(s) in CSC111** – anything goes!

- **Suggested timing**:
  - 5 minutes: generate ideas
  - 5 minutes: make a plan
  - 20 minutes: implement and test

Discussion

What did you
come up with?