

Introduction to Programming (CS 101)

Spring 2024



Lecture 1:

Introduction; Course Administration
Start coding with simple programs

Instructor: Preethi Jyothi

Based on material developed by Prof. Abhiram Ranade and Prof. Manoj Prabhakaran

Computers

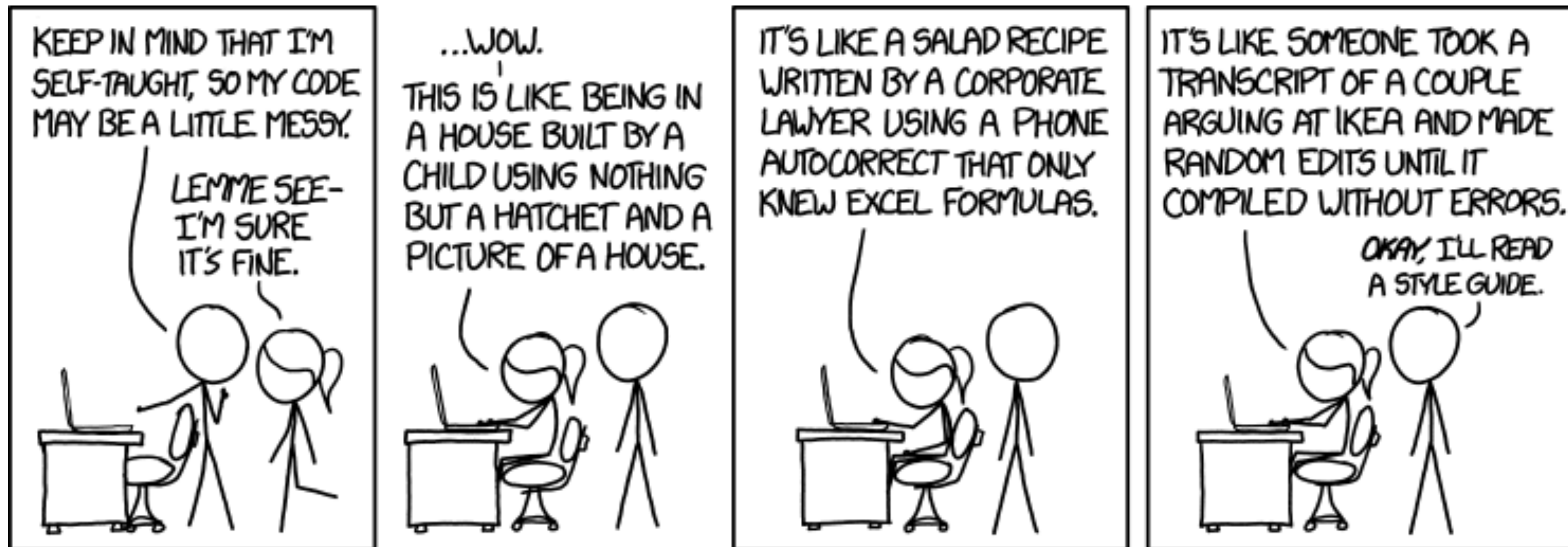
- Computers are an integral part of our daily lives:
 - *Cyber-physical systems* in smart grids, medical device monitoring, ...
 - *Embedded systems* inside your smart phones, TVs, home appliances, ...
 - *Computer systems* grouped together in large networks: Internet, email, search, etc.



- Computers are machines that can be instructed (via **programs** or **code**) to carry out a wide variety of tasks

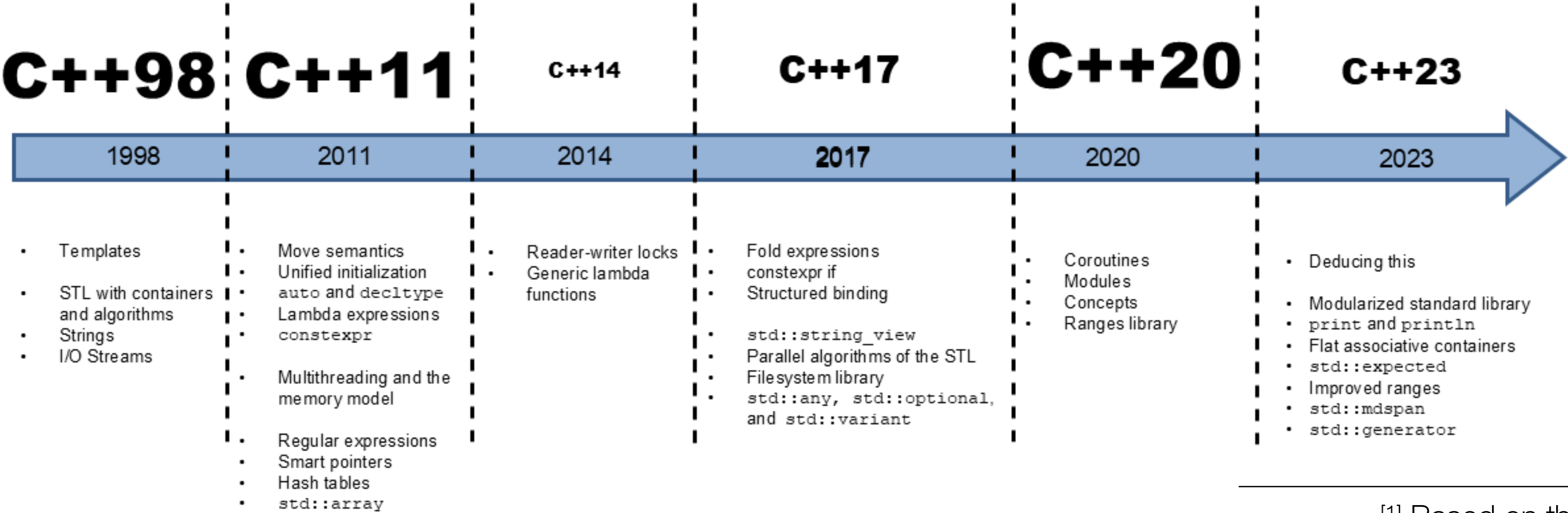
What will this course teach you?

- **Computer programming:** Involves writing instructions in a specific computer language for a computer to follow
- Ability to *computationally think* about problems
- Good coding/programming practices



Programming language for CS101: C++

- **C** Programming Language (created in 1970s by Dennis Ritchie)
 - Developed to build the Unix operating system
 - Consistently among the top four programming languages^[1]
- **C++** Programming Language (created in 1980s by Bjarne Stroustrup)
 - Originally an extension of C, and significantly expanded over time
 - Standards: C++98, C++03, C++11, C++14, C++17, C++20, C++23, ...



[1] Based on the TIOBE index: https://en.wikipedia.org/wiki/TIOBE_index
Image from: <https://www.modernesccpp.com/index.php/c26-the-next-c-standard/>

Programming language for CS101: C++

- **C** Programming Language (created in 1970s by Dennis Ritchie)
 - Developed to build the Unix operating system
 - Consistently among the top four programming languages^[1]
- **C++** Programming Language (created in 1980s by Bjarne Stroustrup)
 - Originally an extension of C, and significantly expanded over time
 - Standards: C++98, C++03, C++11, C++14, C++17, C++20, C++23, ...
- In this course, we will learn a subset of very important features of **C++**

^[1] Based on the TIOBE index: https://en.wikipedia.org/wiki/TIOBE_index

Programming environment

- **Initial Weeks:** C++ augmented with `simplecpp`
- `simplecpp` is a C++ library developed at IITB by Prof. Abhiram Ranade
- Convenient for beginners to programming:
 - Easy-to-use wrappers to simplify syntax
 - Graphics programming -- more visual, more fun!
- Download this file to set up `simplecpp`: <https://drive.google.com/file/d/1nJMm1KwIXkqAKHIVDi-Xt3HrImMJM7Tx/view?usp=sharing> (Will also be posted on Moodle/Bodhitree)
- **Later Weeks:** Transition to C++

How do you get good at programming?

"The only way to learn a new programming language is by writing programs in it"

- Dennis Ritchie, Creator of the *C programming language*, Turing Award (1983)

- You will see your first program in class today.
- Seeing programs is necessary but not sufficient.
- In your first lab, you will set up your system and start coding!



Course Administration and Trivia

CS 101, 2025

Course personnel

- **Instructor:** Preethi Jyothi (Associate Professor, CSE@IITB, <https://www.cse.iitb.ac.in/~pjyothi>)
- **Course Manager:** Firuza Nageshrau Karmali (firuza@cse.iitb.ac.in)
- **CS101 RAs:** Sanchit Kadwe (24M0836), Mayur Pokharkar (24M0840)
- **TAs:** Large ≈ 100 strong team of TAs! Visit <https://www.cse.iitb.ac.in/~cs101/team.html>

Course logistics

CS101 every week: Two lectures (3 hours), One lab (2 hours)

Lectures: Posted on Moodle right before/after class. Also, additional reading material (at times)

Course Website: <https://www.cse.iitb.ac.in/~cs101/>

Attendance: Follow institute policy and use SAFE. TAs will mark attendance in labs.

Bodhitree: Learning management system. All lab-related updates, submissions, exams.

Class Announcements: Made via Moodle or Bodhitree. **Please check your emails.**

Asynchronous Q&A: Approach your TAs or the course manager (Firuza, firuza@cse.iitb.ac.in)

Extra Help Sessions: Schedule up on <https://www.cse.iitb.ac.in/~cs101/>

Course resources

Textbook: "An Introduction to Programming through C++",
Abhiram Ranade, McGraw Hill Education, 2014.

Website: <https://www.cse.iitb.ac.in/~ranade/book.html>

Available in physical and online bookstores.

Integrated with use of **simplecpp**

Course Material: My slides, sample programs, links to additional reading (all on Moodle)

Online Course: Prof. Abhiram Ranade's NPTEL course

<https://archive.nptel.ac.in/noc/courses/noc21/SEM1/noc21-cs38/>

Evaluation (with schedule)

Theory Quiz 1	[Feb 5, 2025]	(7%)
Lab Quiz 1	[Feb 15-16, 2025]	(13%)
Midsem Exam	[Feb 22 - Mar 2, 2025]	(20%)
Theory Quiz 2	[Mar 26, 2025]	(7%)
Lab Quiz 2	[Apr 12-13, 2025]	(13%)
Endsem Exam	[Apr 21 - May 1, 2025]	(30%)
Participation	[Weekly labs]	(10%)

- Last date for showing answer sheets: **May 5, 2025**
- Plan your travel **after May 5, 2025**
- Make-up tests will be offered only for medical reasons. (Show doctor's note to avail of this option.) This will be a mix of theory/lab, held after the endsem exam.

Academic integrity

Code of conduct:

Abide by the following honour code.

- Do not copy, give or receive code during quizzes and exams.
- Attempt lab exercises on your own. You can consult TAs, classmates, reference material, but work on the submissions by yourself. Very useful to identify gaps in understanding.
- If you need help, contact the personnel! (Recall: Extra help sessions.)
- If caught for copying or plagiarism, name of both parties will be handed over to the Department Disciplinary Action Committee (DDAC)¹.

¹<http://www1.iitb.ac.in/newacadhome/punishments201521July.pdf>



Let's Get Coding!
CS 101, 2025



First program: Drawing pictures

- First few programs you will draw pictures on the screen
- Use "Turtle Simulator" contained in **simplecpp**
 - Based on **Logo**: A language invented for teaching programming to children by Seymour Papert et al.
 - Give commands to a turtle to move around.
 - To move the turtle, you write a C++ program.
 - Turtle has a pen, so it draws as it moves.

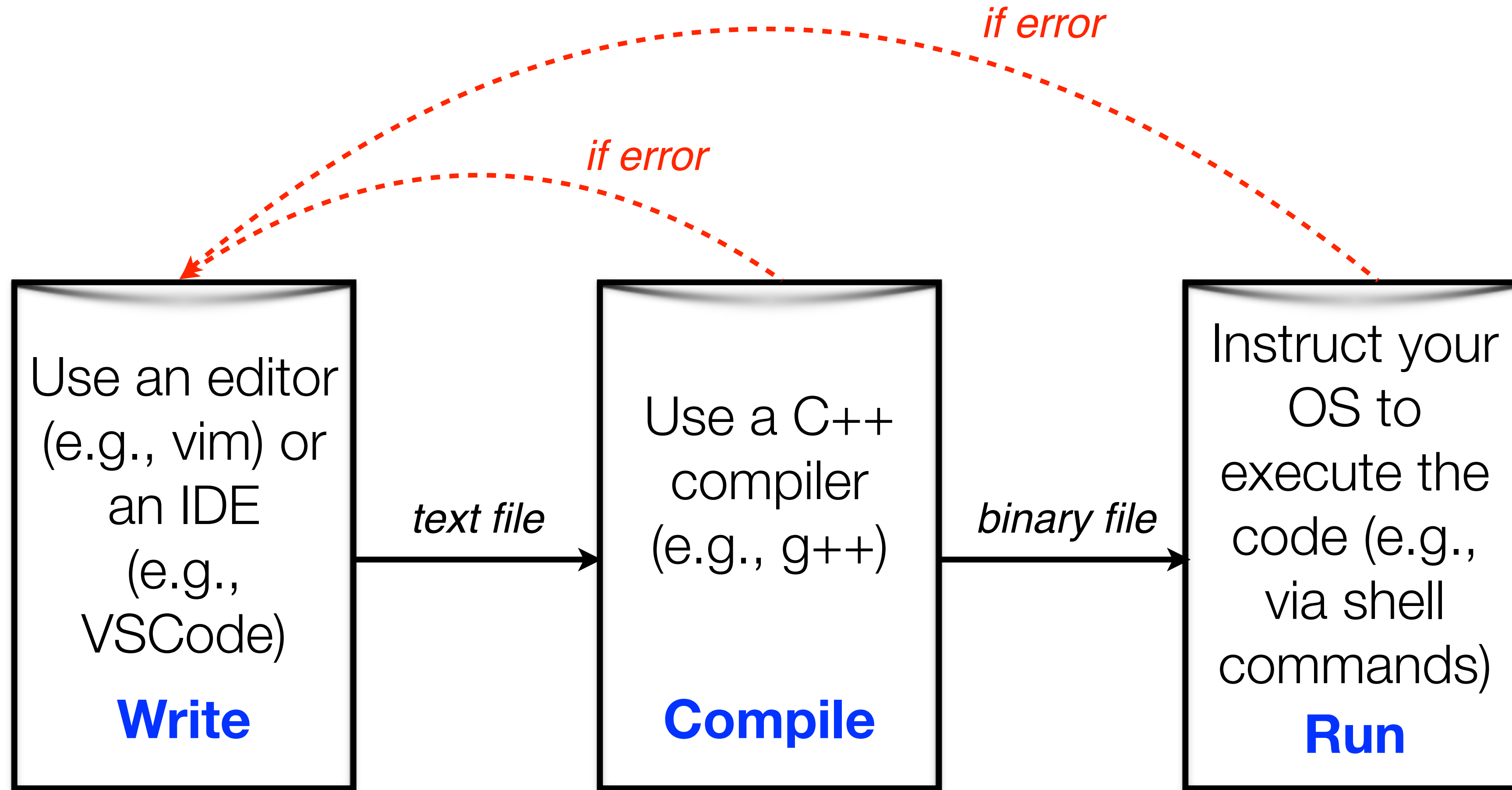
The first program: Draw a square

```
#include <simplecpp>
main_program {
    turtleSim();
    forward(100); right(90);
    forward(100); right(90);
    forward(100); right(90);
    forward(100);
    getClick();
}
```

90 is an argument

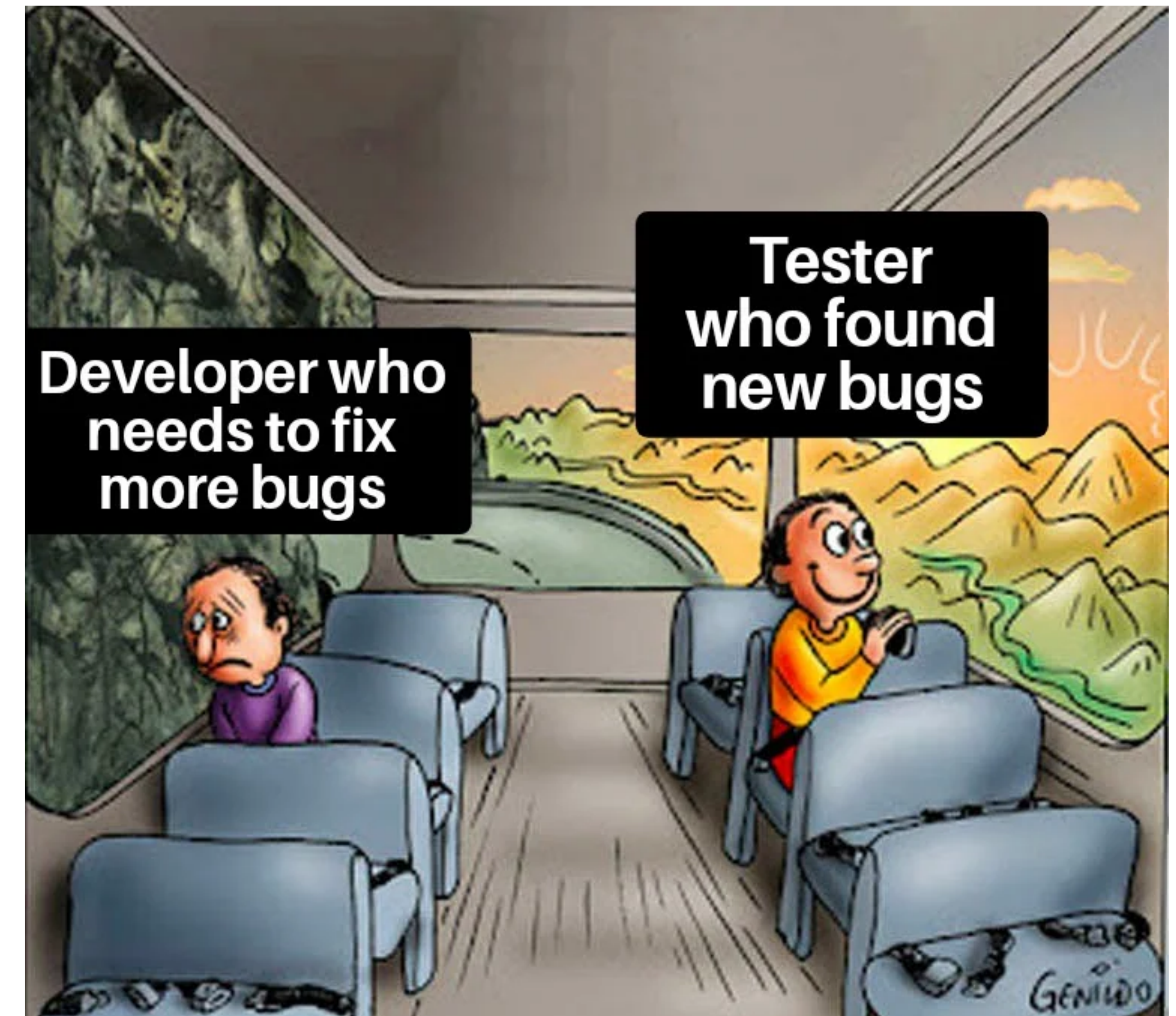
- Use `simplecpp` in the first few lectures; invoked via `#include <simplecpp>`
- Main program starts with `main_program` and enclosed within `{ }`
- `turtleSim()`: Starts the turtle simulator. Opens a window and a turtle at centre facing right
- `forward(n)`: Move the turtle `n` pixels in the direction it is currently facing
- `right(k)`: Turn the turtle `k` degrees to the right
- Note: Semicolons, used as punctuators in C++
- `getClick()`: Wait for a click

How to run the program?



Errors or Bugs 🐛

- A "**bug**" is a defect in the program (either in its syntax or logic)
- Common bugs include syntax errors (e.g., missing semicolon), runtime errors (e.g., divide by zero), logical errors
- **Debugging**: Process of finding and fixing bugs in your code



Errors or Bugs 🐛

- A "**bug**" is a defect in the program (either in its syntax or logic)
- Common bugs include syntax errors (e.g., missing semicolon), runtime errors (e.g., divide by zero), logical errors
- **Debugging**: Process of finding and fixing bugs in your code
- Small bugs can create havoc in large systems
 - Recent example: Crowdstrike bug ^[1]



^[1] https://en.wikipedia.org/wiki/2024_CrowdStrike-related_IT_outages

Redundancy in code?

```
#include <simplecpp>
main_program {
    turtleSim();
    forward(100); right(90);
    forward(100); right(90);
    forward(100); right(90);
    forward(100);
    getClick();
}
```

Use the repeat statement

```
repeat(n) {  
  ⋮  
  body  
  ⋮  
}
```

} **body** consists of one or more statements

- What does this block of code do? **body** is executed **n** times
- This is a "**loop**" (other loop structures will be covered later on)
- Each execution of **body** within a loop is called an "**iteration**"

Draw a square using repeat

"Indentation" {

```
#include <simplecpp>
main_program {
    .....▶ turtleSim();
    .....▶ repeat(4) {
        .....▶ forward(100);
        .....▶ right(90);
    .....▶ }
    .....▶ getClick();
}
```

- Indentation, with appropriate spaces, makes the code readable
- You can use 2, 3 or 4 spaces, but be consistent with this usage
- More about indentation later, as you learn more about conditional blocks, loops, functions, etc.

More `simplecpp` commands

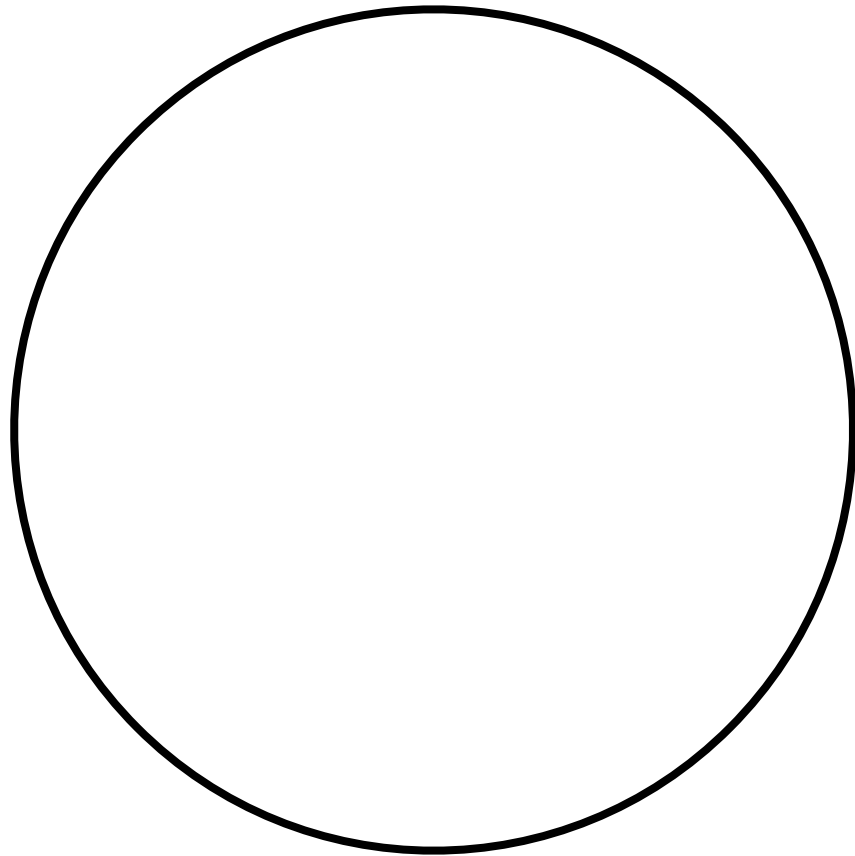
- **`left(d)`**: Turn left **`d`** degrees; equivalent to **`right(-d)`**
- **`penUp()`**, **`penDown()`**: Pen is raised or lowered, respectively. To get the turtle to draw as it moves, the pen should be lowered.
- **`hide()`**: Pen visual (a triangle) is hidden
- **`sqrt(x)`**: Square root of `x`
- **`sine(x)`**, **`cosine(x)`**, **`tangent(x)`**: **`x`** should be in degrees
- Refer to the book for more commands

Code to draw a hexagon, polygon, dotted line

Demo in class and code shared on Moodle

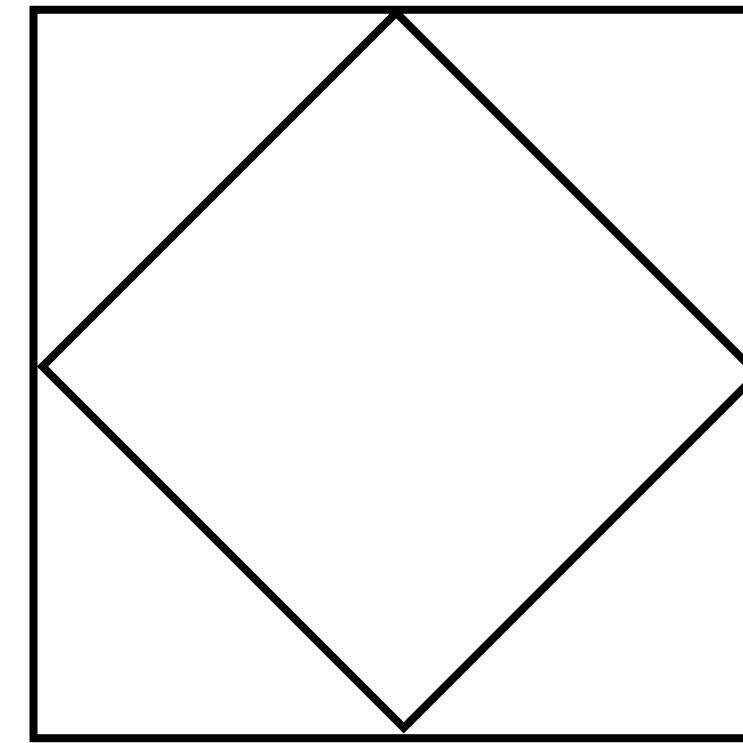
Exercises to try on your own

Draw a circle.



Hint: Draw a regular polygon with a large number of sides

Draw a square of side 100 and inscribe a square inside it.



Hint: Use Pythagoras' theorem to find the length of the inscribed square



Next class: Variables, Operators, Data types
CS 101, 2025