Introduction to C

BU Summer Challenge
Anton Njavro
What have we done last time?

- Introduction to Scratch.
- Basic algorithms.
- Programming concepts.
History of C

- Developed in the early 1970s.
- Its development is closely tied to development of UNIX.
- Its predecessor was interestingly named B.
Significance of C

- Allowed us to think above assembly language.
- Makes systems more portable.
- Crucial in the development of modern software infrastructure.
- Still dominant and omnipresent.
Setting up the Environment

- We will use online GDB environment:
  - https://www.onlinegdb.com/online_c_compiler
Hello, World!
Syntax and Structure of a C Program

- Headers: `#include <stdio.h>`
- `main()` function
- Statements ending with `;`
- Comments: `//` or `/* ... */`
- Escape sequences: `\n, \t, \\, \"`
Compilation Pipeline

1. Preprocessing:
   a. Handles directives such as include, define, macro
   b. Produces expanded source file.

2. Compilation:
   a. Converts source to assembly.

3. Assembly:
   a. Converts assembly to machine code.
   b. Results in object file.

4. Linking:
   a. Combines object files and libraries into executable files.

- Important to understand for debugging and optimization.
Data Types and Variables

- **int**: Integer type, typically 4 bytes.
- **float**: Single-precision floating-point, typically 4 bytes.
- **double**: Double-precision floating-point, typically 8 bytes.
- **char**: Character type, typically 1 byte.
- **Void** data type, used for functions that do not return value. (Example?)
Operators

- Arithmetic operators.
- Relational operators.
- Logical operators.
- Bitwise operators.
- Assignment operators.
Control Flow Statements

- Conditional statements:
  - If, if-else, nested, switch-case.
- Loops:
  - for loop, while loop, do while loop.
- Break and continue.
<brian kernighan>