AI INSTITUTE #AIISC
UNIVERSITY OF SOUTH CAROLINA

UNIVERSITY OF SOUTH CAROLINA

CSCE 240: Advanced Programming Techniques
Lecture 4: Input and Output, Formatting

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

18TH JANUARY 2024

Carolinian Creed: "I will practice personal and academic integrity."

Credits: Some material reused with permission of Dr. Jeremy Lewis. Others used as cited with thanks.

Organization of Lecture 4

- Introduction Section
  - Recap of Lecture 3
- Main Section
  - Review home assignment #1: (FileBasedCalculator)
    - Peer activity: code review, testing
  - Concept: Handling mixed data types
  - Concept: Printing with formatting
- Concluding Section
  - About next lecture – Lecture 5
  - Ask me anything

Introduction Section

Recap of Lecture 3

- We discussed the concepts of data types, strings
- We discussed the concepts of streams and error handling
- We looked at programs in C++, Java and python on file handling
- Homework assignment - FileBasedCalculator

Main Section

Programming Home Work (#1) – C++

- Write a program called FileBasedCalculator.
  - It reads three lines from an input file (called input.txt): the operation to be done (add, subtract, multiply or divide), and two integer numbers.
  - It writes two lines to an output file (called output.txt). The first line says – ”The result of <operation> on <num1> and <num2> is below”. The second line has the result.
- Modify the program so that it can handle missing input file name.

Peer Review: Homework Assignment #1

1. Go to spread sheet and on ”Homework Assignments - Peer Review” tab
2. Go to the row with your name
3. Peer review (10 mins)
    1. Enter roll number of person on your **LEFT** under “ID of code reviewer”
    2. Share code for the reviewer to see
    3. Reviewer: enter review (1-5)
    4. **Note**: negotiate – review code of neighbor or get own’s code reviewed
4. Peer test (10 mins)
    1. Enter roll number of person on your **RIGHT** under “ID of code tester”
    2. Share command line for the tester to see
    3. Tester: enter review (1-5)
    4. **Note**: negotiate – test code of neighbor or get own’s code tested

Peer Reviewing Guideline (10 mins)

- Look out for
  - Can you understand what the code is doing ?
  - Can you explain the code to someone else (non-coder) ?
  - Can you spot possible issues without running it?
    - Are the variables initialized ?
    - Are files closed?
    - Is their unnecessary code bloat ?
- What not to judge
  - Usage of language features, unless they are inappropriate
- Assign rating
  - 1: code not available
  - 2: code with major issues
  - 3: code with minor issues
  - 4:
  - 5: no issues

Peer Testing Guideline (10 mins)

- Look out for
  - Does the program run as the coder wanted it to be (specification) ?
  - Does the program run as the instructor wanted it to be (requirement - customer) ?
  - Does the program terminate abruptly ?
  - Any special feature?
- What not to judge
  - Person writing the code
- Assign rating
  - 1: code not available
  - 2: code runs with major issues (abnormal termination, incomplete features)
  - 3: code runs with minor issues
  - 4:
  - 5: No issues

Discussion

- Peer Code Reviewing
- Peer Testing

Concept: Handling Mixed Types

We want to distinguish product name, model, specification, year, …

The Company announces new product, service and software offerings at various times during the year. Significant announcements during fiscal year 2023 included the following:

First Quarter 2023:

- iPad and iPad Pro;
- Next-generation Apple TV 4K; and
- MLS Season Pass, a Major League Soccer subscription streaming service.

Second Quarter 2023:

- MacBook Pro 14", MacBook Pro 16" and Mac mini; and
- Second-generation HomePod.

Third Quarter 2023:

- MacBook Air 15", Mac Studio and Mac Pro;
- Apple Vision Pro™, the Company's first spatial computer featuring its new visionOS™, expected to be available in early calendar year 2024; and
- iOS 17, macOS Sonoma, iPadOS 17, tvOS 17 and watchOS 10, updates to the Company's operating systems.

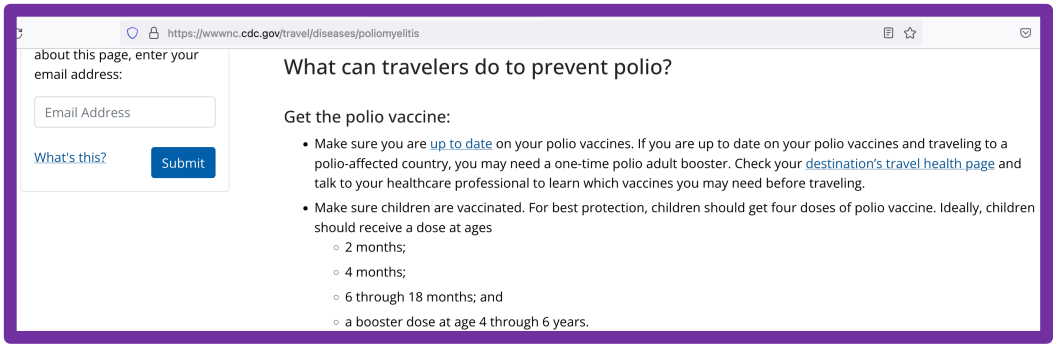
Fourth Quarter 2023:

- iPhone 15, iPhone 15 Plus, iPhone 15 Pro and iPhone 15 Pro Max; and
- Apple Watch Series 9 and Apple Watch Ultra 2.

From Apple's 10-K: 10-k: https://www.sec.gov/ix?doc=/Archives/edgar/data/320193/000032019323000106/aapl-20230930.htm#i1cb1ba018cb1455aa66bd3f9ab0c5b1a_175

Concept: Handling Mixed Types

Make sure children are vaccinated. For best protection, children should get four doses of polio vaccine. Ideally, children should receive a dose at ages 2 months;
•4 months;
•6 through 18 months; and
•a booster dose at age 4 through 6 years.



https://wwwnc.cdc.gov/travel/diseases/poliomyelitis

Concept: Handling Mixed Types

District 66
Photo of Representative Gilda Cobb-Hunter
Representative Gilda Cobb-Hunter (D)
4188 Five Chop Road, Orangeburg 29115

Concept: Data Types

Common C++ types

Type	Typical Bit Width	Typical Range
char	1byte	-127 to 127 or 0 to 255
unsigned char	1byte	0 to 255
signed char	1byte	-127 to 127
int	4bytes	-2147483648 to 2147483647
unsigned int	4bytes	0 to 4294967295
signed int	4bytes	-2147483648 to 2147483647
short int	2bytes	-32768 to 32767
unsigned short int	2bytes	0 to 65,535
signed short int	2bytes	-32768 to 32767
long int	8bytes	-2,147,483,648 to 2,147,483,647
signed long int	8bytes	same as long int
unsigned long int	8bytes	0 to 4,294,967,295
long long int	8bytes	-(2^63) to (2^63)-1
unsigned long long int	8bytes	0 to 18,446,744,073,709,551,615
float	4bytes	
double	8bytes	
long double	12bytes	
wchar_t	2 or 4 bytes	1 wide character

Credit and Reference: https://www.tutorialspoint.com/cplusplus/cpp_data_types.htm

Mixed Data Types

- Examples:
  - Char, string, int, double on the same line
  - Char, string, int, float on different lines
  - Both
- Strategy
  - Read as characters/ strings by line
  - Parse each line
- Assumption
  - Reader has idea of what data type is at a location
- *What if the reader cannot assume?*

https://www.scstatehouse.gov/member.php?chamber=H

District 66
Photo of Representative Gilda Cobb-Hunter
Representative Gilda Cobb-Hunter (D)
4188 Five Chop Road, Orangeburg 29115

Code Demo

- **Function:** demoReadMixedFile()
  - Once we have each word, we can convert to specific format based on our expectation of data type expected at that position
  - Github: https://github.com/biplav-s/course-adv-proglang/blob/main/sample-code/CandC%2B%2B/Class3and4_C%2B%2B_IO/src/Class3and4_C%2B%2B_IO.cpp

- Question: *What if the reader cannot assume?*

Type Conversions

- Widening (promotion)

From	To
Any `signed` or `unsigned` integral type except `long long` or `__int64`	`double`
`bool` or `char`	Any other built-in type
`short` or `wchar_t`	`int`, `long`, `long long`
`int`, `long`	`long long`
`float`	`double`

- Narrowing conversions (coercion)

Reference and figure credit: https://docs.microsoft.com/en-us/cpp/cpp/type-conversions-and-type-safety-modern-cpp?view=msvc-170

Discussion: Sorting of Characters

- Setting
  - Input: [‘a’, ‘z’, ‘i’]
  - Output: [‘a’, ‘i’, ‘z’]

- Question: can we reuse previous sorting program with minimal change?
  - If yes, how ?

Concept: Formatted Printing

Concept: Error Handling

%[flags][width][.precision][length]**specifier**

specifier	**Output**	**Example**
d *or* i	Signed decimal integer	392
u	Unsigned decimal integer	7235
o	Unsigned octal	610
x	Unsigned hexadecimal integer	7fa
X	Unsigned hexadecimal integer (uppercase)	7FA
f	Decimal floating point, lowercase	392.65
F	Decimal floating point, uppercase	392.65
e	Scientific notation (mantissa/exponent), lowercase	3.9265e+2
E	Scientific notation (mantissa/exponent), uppercase	3.9265E+2
g	Use the shortest representation: %e or %f	392.65
G	Use the shortest representation: %E or %F	392.65
a	Hexadecimal floating point, lowercase	-0xc.90fep-2
A	Hexadecimal floating point, uppercase	-0XC.90FEP-2
c	Character	a
s	String of characters	sample
p	Pointer address	b8000000
n	Nothing printed. The corresponding argument must be a pointer to a signed int. The number of characters written so far is stored in the pointed location.	
%	A % followed by another % character will write a single % to the stream.	%

Reference: Content courtesy - https://www.cplusplus.com/reference/cstdio/printf/

Code Demo

- **Function:** demoFormattedPrinting()
  - C's printf / sprintf allows fine-grained control and data type specific
  - Has inspired formatting support in other languages

Discussion: Course Project

Course Project – Knowing About Companies

- **Project**: Develop collaborative assistants (chatbots) that offer useful information about companies
- Specifically, use the EDGAR dataset on companies at: https://www.sec.gov/edgar/searchedgar/companysearch.
  - For Apple, it is: https://www.sec.gov/edgar/browse/?CIK=320193&owner=exclude
- **Each student will choose two companies (from thousand available).**
- Programming assignment programs will: (1) extract data about two companies from 10-k, (2) process it, (3) make content available in a command-line interface, (4) handle any user query and (5) report on interaction statistics.

Discussion: Nature and Simplifications

- Once you select a company, the content is also fixed.
  - Enter selection in column F of spreadsheet
- Some simplifications
  - **Download local copy** v/s web query
  - **Read static content first**
  - **Handle a subset of content**
  - **Have default handling for questions** the chatbot does not understand
- Do project in a language you are most comfortable with
- Use all advanced programming concepts to simplify coding

Discussion: Chatbot Loop

- Input: from user (called utterance)
  - Problem specific query (i.e., about company risk factors)
  - Chitchat
  - Unrelated
- Output: from system (response)
  - Handle unrelated
  - Handle chitchat
  - Answer to query

- **Do it until user say over!**

ToDo

- Identify two companies you want to focus on
- Access their 10-K report from EDGAR dataset on companies. Use search at: https://www.sec.gov/edgar/searchedgar/companysearch
- List the companies in spreadsheet

Concluding Section

Lecture 4: Concluding Comments

- We experienced peer review on home works
- Discussed the concepts of mixed types
- Discussed formatted printing

About Next Lecture – Lecture 5

Lecture 5: Memory Management

- Memory management
  - Dynamic object creation
  - Object destruction
- User defined types