AI INSTITUTE #AIISC
UNIVERSITY OF SOUTH CAROLINA

UNIVERSITY OF SOUTH CAROLINA

CSCE 240: Advanced Programming Techniques

Lecture 5: User-defined Types, Memory Management,
HW 2 (Given), Prog 1 (Start)

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

25TH JANUARY 2022

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 5

- Introduction Section
  - Recap of Lecture 4
  - TA and SI Updates
- Main Section
  - Concept: User defined types
  - Concept: Memory management
  - Home work2 – given
  - Programming assignment 1 - Start
- Concluding Section
  - About next lecture – Lecture 6
  - Ask me anything

Introduction Section

Recap of Lecture 4

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

Updates from TA, SU

- TA update: Yuxiang Sun (Cherry)
- SI update: Blake Seekings

Main Section

C/C++ Compilation Process

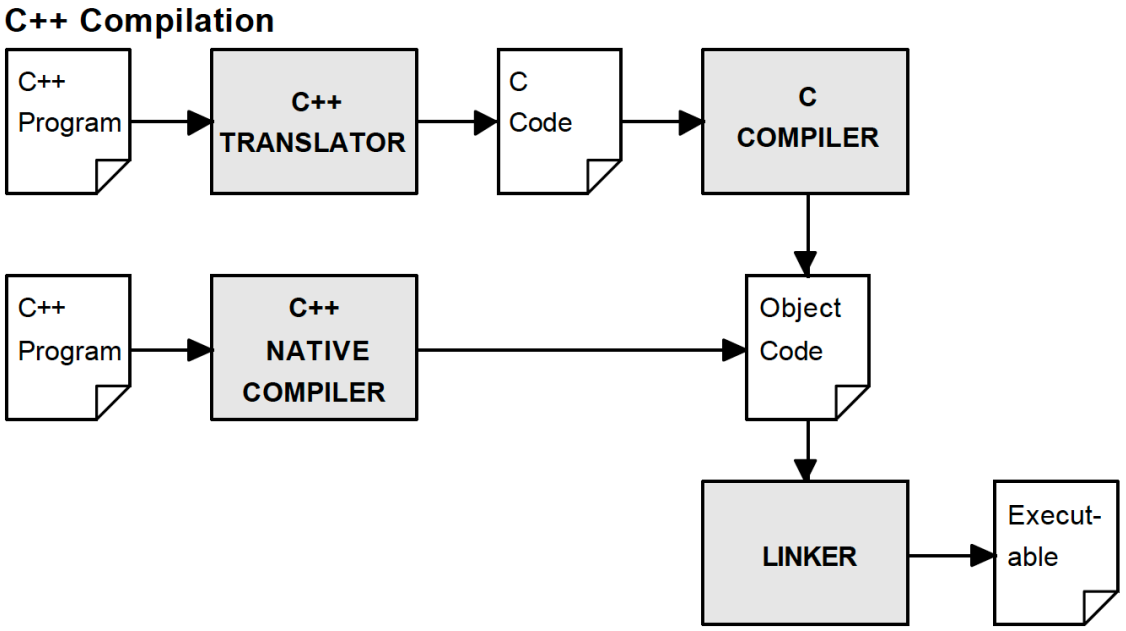

Figure courtesy: C++ Essentials by Sharam Hekmat

Concept: User Defined Types

User Defined Types

- When we start solving real world problems, we often use a set of information together
  - Examples:
    - Name = {title, first-name, middle-name, last-name, suffix}
    - Address = {Street name, Number, City, State, Zip code}
  - Need not be of the same type as a language's pre-determined / basic data types
- May be of the same or different basic types

Why User Defined Types ?

- Ease of writing and maintaining code
  - Abstraction helps in communication
  - Code is easier to understand
  - Code becomes easier to test
- No impact on code's executional performance
- C++ concepts
  - Struct
  - Class (Object Oriented Programming)

Code Illustration

- Problem
  - Represent person details
- Example: **Function:** demoStruct()

Concept: Memory Management

Types of Memory Allocation

- Static: when size is known at compile time
  - Basic data types
  - Arrays
- Dynamic: when size not known ahead of time
  - User defined types
  - Linked list

Advantages and Disadvantages of Dynamic Allocation

- Advantages
  - Does not waste memory
  - Does not ask user for information (size, data type) user may not know
  - Eases supporting of user-defined types
- Dis-advantages
  - Requires developer to be careful with allocation and de-allocation
  - Code is often complex compared to static allocation

Code Illustration

- **Problem**: Get average of a set of random numbers
- Note
  - Numbers not known ahead of time
  - How much to allocate?
    - Too high: wasting space
    - Too low: program will behave incorrectly

Code Illustration

- Static allocation – **Function:** demoStaticMemoryAllocation()

- Dynamic allocation – **Function:** demoDynamicMemoryAllocation()

Home Work 2

Programming Home Work (#2) – C++

- Write a program called GeometricPropertyCalculator.
  - The program reads an input file (called input.txt). Each line in the file contains dimensions of a geometric shape – rectangle, shape and triangle. Specifically:
    - For rectangle, it contains – RECTANGLE <length-in-cm> <breadth-in-cm>
    - For circle, it contains – CIRCLE <radius-in-cm>
    - For triangle, it contains – TRIANGLE <side-1-in-cm> <side-2-in-cm> <side-3-in-cm>
  - The user specifies the property to calculate as argument to the program: 1 for AREA and 2 for PERIMETER
  - The program writes output lines to an output file (called output.txt) for each shape that it reads and the property – AREA or PERIMETER.
    - For example, for RECTANGLE and property as AREA, the program should write – RECTANGLE AREA <calculated value>
  - Write GeometricPropertyCalculator in C++
    - It should support RECTANGLE, CIRCLE and TRIANGLE
    - It should support properties AREA and PERIMETER
    - If there is insufficient information, the program should give an error. E.g. TRIANGLE AREA “Not enough information to calculate”

Programming Home Work (#2) – C++

- Code guidelines
  - Have sub-directories in your folder
    - src sub-folder, (or code) for code
    - data sub-folder, for input.txt and output.txt
    - doc sub-folder, for documentation on what the code does or sample output

- Hint
  - Area
    - Rectangle: length x breadth
    - Circle: pi * r^2
    - Triangle: -
  - Perimeter
    - Rectangle: 2 * (length + breadth)
    - Circle: 2 * pi * r
    - Triangle: sum of sides

Discussion: Programming Assignment # 1 Course Project

Course Project – Assembling of Prog. Assignments

- **Project**: Develop collaborative assistants (chatbots) that offer innovative and ethical solutions to real-world problems ! *(Based on competition - https://sites.google.com/view/casy-2-0-track1/contest)*
- Specifically, **the project will be building a chatbot that can answer questions about a South Carolina member of state legislature from**: https://www.scstatehouse.gov/member.php?chamber=H
  - Each student will choose a district (from 122 available).
  - Programming assignment programs will: (1) extract data from the district, (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 district, the elected legislator is fixed.
- 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 district chosen)
  - Chitchat
  - Unrelated
- Output: from system (response)
  - Handle unrelated
  - Handle chitchat
  - Answer to query
- **Do it until user say over!**

Handling different data types

Show formatted content!

Core Programs Needed for Project

- Prog 1: extract data from the district
- Prog 2: process it based on questions
- Prog 3: make content available in a command-line interface
- Prog 4: handle any user query and
- Prog 5: report statistics on interaction of a session, across session

Programming Assignment # 1

- Goal: extract data from the district of choice
  - Language of choice: Any from the three (C++, Java, Python)
- Program should do the following:
  - Take district name as input
  - Read content from the district's URL or a local text version of the district page
  - Report statistics of content: lines, words, chars
  - Write content out in an output file formatted with indentation
- Code organization
  - Create a folder in your GitHub called "prog1-extractor"
  - Have sub-folders: src (or code), data, doc, test
  - Write a 1-page report in ./doc sub-folder
  - Send a confirmation that code is done to instructor and TA, and update Google sheet

Concluding Section

Lecture 5: Concluding Comments

- We discussed
  - the concept of user-defined types
  - the concepts of static and dynamic memory allocation
- Discussed Home Work 2 (due Thursday, Jan 27)
  - Peer evaluation in class
- Discussed Programming Assignment #1 (due Thursday, Feb 3)

About Next Lecture – Lecture 6

Lecture 6: Object Oriented Concepts

- Home work 2 due
  - Peer evaluation in class
- Concepts: Classes and Objects
- Project: Chatbots Background