AI INSTITUTE #AIISC
UNIVERSITY OF SOUTH CAROLINA

UNIVERSITY OF SOUTH CAROLINA

CSCE 240: Advanced Programming Techniques
Lecture 8: Object Oriented Concepts - Continued

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

2ND FEBRUARY 2023

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 8

- Introduction Section
  - Recap of Lecture 7
- Main Section
  - Concept: Multiple classes – UML diagrams and Object Oriented programs
  - Concept: Code organization
  - Discussion: Prog. Assignment #1 and Project discussion
- Concluding Section
  - About next lecture – Lecture 9
  - Ask me anything

Introduction Section

Recap of Lecture 7

- We introduced UML – a language independent notation for communicating about OO software
- Looked at concept of encapsulation
- Discussed background of chatbot

ACM Hackathon

"The Code-A-Thon is a programming competition we hold each semester with four divisions: 145, 146, 240, and 350. Each number corresponds with the highest CSCE course the students are enrolled in or have most recently completed. The top three placements in each division will receive $100, $75, and $50 respectively. The mission of the competition is to encourage algorithmic problem-solving and help expose the contestants to solving real-world problems including (but not limited to) pathfinding, graph traversal, binary search, flood fill, backtracking, and dynamic programming.

We will be having the kickoff at 6:30 pm in WIRED Cafe on February 24th. The competition will start at 7:00 pm that same day and run until 7:00 pm on Feb. 25th. For any students who are interested, I am attaching a flyer (in both pptx and pdf format) if don't mind sharing that. Aside from that, we will post all information on how to register and compete on our website https://acm.cse.sc.edu/"

ACM Student Chapter, USC

Main Section

Concept: Working with Multiple Classes

Example: Refrigerator

- is a type of electrical appliance [Generalization/ Specialization]
- has freezer section, cooler section, water compartment [Composition]
- can have ice tray, food items [Aggregation]

Example: Refrigerator in UML Classes

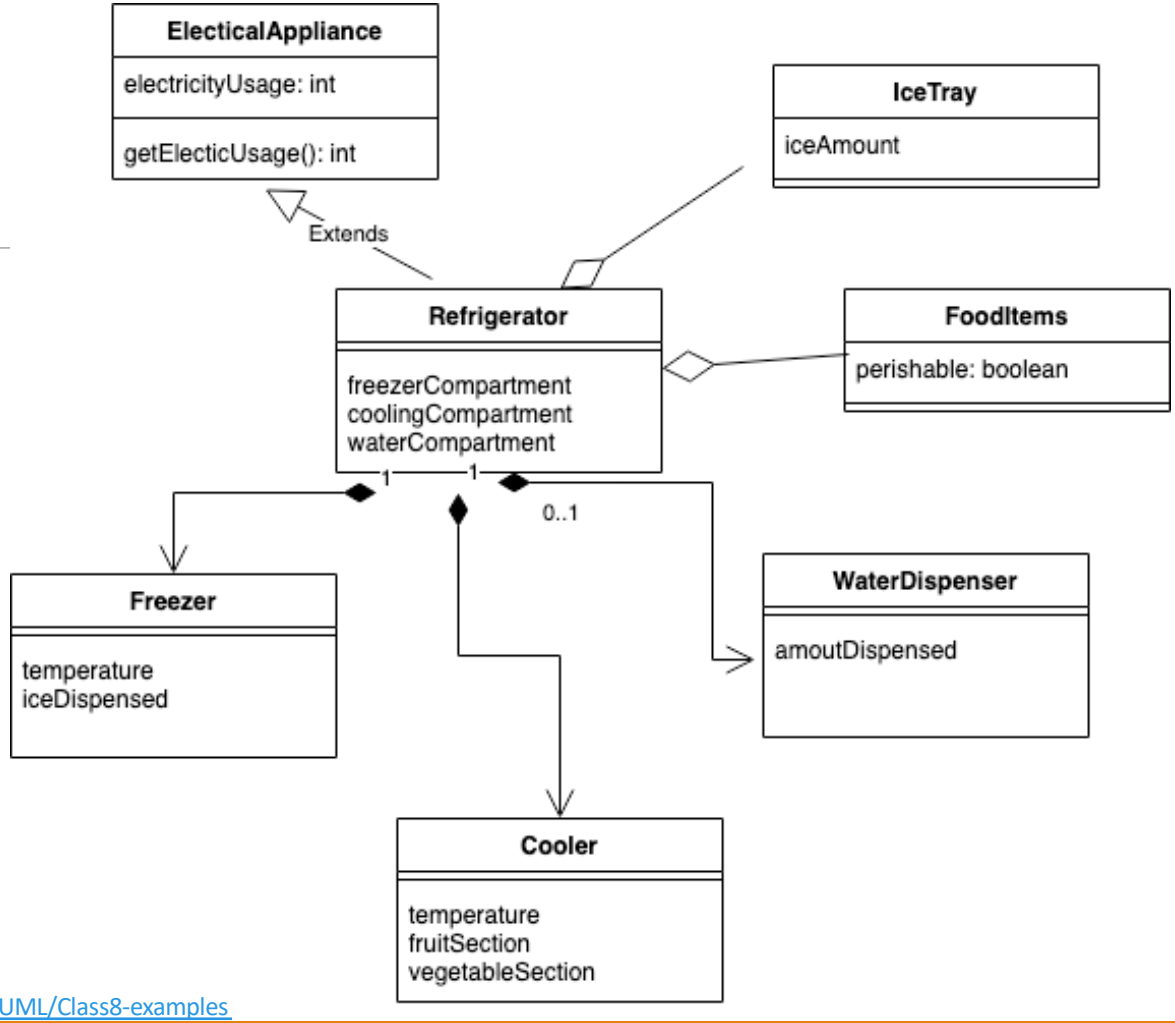


Diagram at:
https://github.com/biplav-s/course-adv-proglang/tree/main/sample-code/UML/Class8-examples

Relationship Types

- Association: is related to
- Generalization: is a special type of (inverse relationship: specialization of)
- Aggregation: is made up of, but can also exist independently
- Composition: is made up of, but cannot exist independently

References:

1. UML 2.5.1 specs
2. Tutorial: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/

Example: Representative Information

- Contact Information (Type-I1)
  - Name
  - Label: District, County, Party
  - Addresses: Columbia, Home
  - Phone: Business, Home
- Personal Information (Type-I2)
- Committee Assignments (Type-I3)
- Sponsored Bills in the House (Type-I4)
- Voting Record (Type-I5)

Modeling Questions

- Consider: **Contact Information (Type-I1)**
  - Name
  - Label: District, County, **Party**
  - Addresses: Columbia, Home
  - Phone: **Business**, Home
- Composition or aggregation
  - If reusing information, model as aggregation (e.g., Party)
  - If specific to the class, model as composition (e.g., Business phone number)

Example: Representative Information

- Contact Information (Type-I1)
  - Name
  - Region
  - Addresses: Columbia, Home
  - Phone: Business, Home
- Personal Information (Type-I2)
- Committee Assignments (Type-I3)
- Sponsored Bills in the House (Type-I4)
- Voting Record (Type-I5)
- **Service in Public Office (Type-I6)**

Representative Terry Alexander

Democrat - Florence
District 59 - Darlington & Florence Counties - **Map**

Columbia Address
314C Blatt Bldg.
Columbia 29201

Home Address
1646 Harris Court
Florence 29501

Business Phone (803) 734-3004

Home Phone (843) 665-7321

Send message to Representative Alexander

Personal Information

- Education Consultant & Pastor
- Residing at 1646 Harris Court, Florence
- Born January 23, 1955 in Florence
- Son of the late James and Adell Alexander
- Durham Business College, A.D., 1976
- Francis Marion University, B.A., 1991
- Howard University School of Divinity, M. Div., 1998
- Married to Starlee Davis Alexander, 2 children, Terrell McClain and Matthew
- Pastor, Wayside Chapel Baptist Church
- Career Development Consultant
- Adjunct Professor of Religion, Limestone College
- Pee Dee Regional Council of Governments
- Past President, Habitat for Humanity, Board of Directors
- Charter member, The Florence Breakfast Rotary Club
- Past President, Boys and Girls Club of Florence
- Boy Scouts of the Pee Dee Executive Boards
- Florence Branch, NAACP, past President
- Mercy Medicine Board
- Pee Dee Chapter American Red Cross
- 100 Black Men of the Pee Dee
- Kappa Alpha Psi Fraternity, Inc.
- Francis Marion Society
- National Association of County Officials
- National Association of Black County Officials
- South Carolina Association of Black County Officials
- South Carolina Association of Guidance Counselors
- South Carolina Alliance of Black Educators

Committee Assignments

- **Education and Public Works, *2nd V.C.***
- **Regulations and Admin. Procedures**

Sponsored Bills in the House

- Primary Sponsor: (•) Yes ( ) No
- Search Session: 2021-2022 (124) Find Bills

Voting Record

- Search Session: 2021-2022 (124) Find Votes

Service In Public Office

- Florence County Council, 1990-06, District Number 3
- House of Representatives, 2007 - Present

Example: Disease

- What (is the disease)?
  - What are they types?
  - What causes it?
  - What are the symptoms?
  - What should one do to treat the disease ?
- Who is affected?
  - Who is at risk?
- How is the disease diagnosed
- When to call doctor?
- More information
  - After travel

S1: https://www.cdc.gov/travel/diseases/malaria

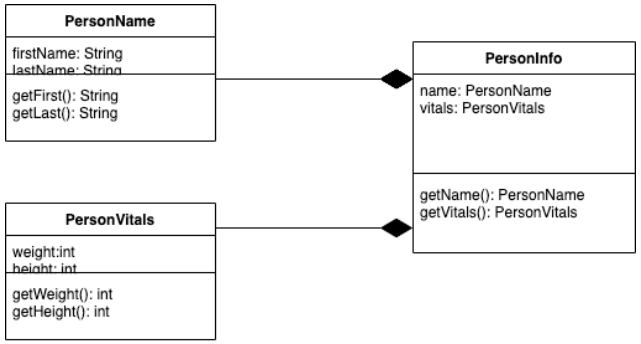
- What is malaria?
- Who is at risk?
- What can travelers do to prevent malaria?
- After Travel
- More Information

S2: https://www.webmd.com/a-to-z-guides/malaria-symptoms

- What Is Malaria?
- Malaria Causes and Risk Factors
- Types of Malaria
- Symptoms
- When to Call a Doctor About Malaria
- Malaria Diagnosis
- Malaria Treatment
- Malaria Complications
- Malaria Vaccine

Exercise

- Using browser, go to: https://app.diagrams.net/
- Go to: File -> Open from -> Device -> and load file “Example.drawio”
  (From: https://github.com/biplav-s/course-adv-proglang/tree/main/sample-code/UML/Class7-examples)
- Extend it to represent
  Contact Information (Type-I1) // Sub-type of PersonInfo
  - Name // Type: String or reuse PersonName
  - Label: District, County, Party
  - Addresses: Columbia, Home
  - Phone: Business, Home
- Review it
  - Does your chosen district’s (59 or 104) representative have weight/ height information on the website ? If so, we can already handle it !!
- You can save file or export the diagram in any supported format



Concept: Code Organization

Options to Organize Code

- Option 1: All classes in the same file
  - Not suggested
  - Makes code hard to reuse
  - Makes code hard to understand
- Option 2: Have classes as separate header (.h) and implementation files (.cpp)
  - Each class should have a clear purpose
  - Header file has specification of the class: data members and function specification, but not implementation; implementation is in .cpp
  - Good suggestions in: http://websites.umich.edu/~eecs381/handouts/CppHeaderFileGuidelines.pdf
  - Someone reusing the code will only need to look at the header file, not implementation
  - **Pitfall**: too many files if the project is small, especially if development is by a single person team
- Option 3: Mix and match of above
  - Separate classes for important concepts
  - Utility "class" for rest of the data members, functions

Review: Code Implementation in Instructor Code

- As supported by Eclipse IDE !
  - Creates header and implementation files automatically
  - sub-directories with clear purpose
- Keeping project size small

Code Review

- PersonName: has separate header and implementation files for a class
- SimpleRelational: has single class implementation file
- Class7and8_C++_OO: is a utility file

Discussion: Course Project

Course Project – Building and Assembling of Prog. Assignments in Health

- **Project**: Develop collaborative assistants (chatbots) that offer useful information about diseases
- Specifically, use the CDC dataset on diseases at: https://wwwnc.cdc.gov/travel/diseases
  - For polio, it is: https://wwwnc.cdc.gov/travel/diseases/poliomyelitis
  - Each student will choose two diseases (from 47 available).
  - Each student will also use data about the disease from WebMD. Example for polio - https://www.webmd.com/children/what-is-polio
  - Programming assignment programs will: (1) extract data about a disease from two sites, (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 disease, the scope of answers 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

Suggested Scope is a Drastic Simplification

- **Users**: 1
- **Modality**: text
- **Data:** static
  (no data in S1 and S2 is dynamic)
- **Personalization**: none
- **Form**: command line
- **Purpose**: information provider
- **Domain**: specific to disease chosen

Core Programs Needed for Project

- Prog 1: extract data from the disease pages
- 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 disease of choice
  - Language of choice: Any from the three (C++, Java, Python)
- Program should do the following:
  - Take disease as input
  - Read content about the disease
    - from the disease's URL from CDC and WebMD, OR
    - a local text version of the disease pages // Keep them as separate files with names <disease>-<source>.txt
  - 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 update Google sheet

PA: Code Testing Rubric Used

- 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 ?
  - Is there a hardcoding of directory ? Paths should be relative to code base directory.
  - Any special feature?
- What not to judge
  - Length of documentation. It can just be short and accurate.
  - Person writing the code

Assign rating (out of 100 -/+)
- -100: code not available
- -80: code with major issues (e.g., abnormal termination, incomplete features)
- -60: code with minor issues
- -20:
- (full marks): no issues
- +20: special features

Discussion

Concluding Section

Lecture 8: Concluding Comments

- We relooked at relationships between classes
- We discussed code organization
- We discussed Prog. Assignment #1 due today

About Next Lecture – Lecture 9

Lecture 9: Object Oriented - Inheritance

- OO - Inheritance
- Home work 3 will be given
- Programming assignment #2 begins – hints given

9	Feb 7 (Tu)	OO – inheritance	Prog 2 - start
10	Feb 9 (Th)	Regex, OO - polymorphism	Remote
11	Feb 14 (Tu)	In class test	Quiz 1 – Remote, due end of day
12	Feb 16 (Th)	Review: inheritance, Polymorphism	HW 3 due