CSCE 240: Advanced Programming Techniques
Lecture 14: Constructors and Destructors

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

24TH FEBRUARY 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 14

- Introduction Section
  - Recap of Lecture 13
  - TA and SI Updates
- Main Section
  - Concept: Constructors
  - Concept: Destructors
  - Home work #4
  - Discussion: Project, Programming Assignment #3
- Concluding Section
  - About next lecture – Lecture 15
  - Ask me anything

Introduction Section

Recap of Lecture 13

- Looked at Errors
- Looked at Exception Handling
- Examples of Exceptions
  - In C++, Java, Python
  - Creating new exception handlers in C++

Announcements

- Chatbots – Event on March 18, 2022
  - Collaborative Assistants for Society (CASY) – in person and virtual event on campus
  - 9:30 am – 1:00 pm; talks and student use-cases

- Details and registration info: https://casy.aiisc.ai

Updates from TA, SU

- TA update: Yuxiang Sun (Cherry)
  - HW3 marks now on Blackboard

- SI update: Blake Seekings

Main Section

Concept: Constructors

Constructor - What is It?

- Special function in every class
  - Always has the same name as the class itself
  - Does not have an explicit return type
  - Multiple constructors possible per class

- **Purpose**: Used to initialize objects of that class

Specification

```
class PersonName {
        string firstName;
        string lastName;

public:
        PersonName();
        PersonName(string);
        PersonName(string, string);
…
```

https://github.com/biplav-s/course-adv-proglang/blob/main/sample-code/CandC%2B%2B/Class7and8_C%2B%2B_OO/src/headers/PersonName.h

Constructor - What is It?

- Special function in every class
  - Always has the same name as the class itself
  - Does not have an explicit return type
    Multiple constructors possible per class

- **Purpose**: Used to initialize objects of that class

Usage

```
PersonName p1;
PersonName p2("Joginder");
PersonName p3("Joginder", "Singh"
```

Implementation

```
PersonName::PersonName() {
        firstName = "default-Maria";
        lastName = "default-Wang";
}

PersonName::PersonName(string first) {
        firstName = first;
        lastName = "default-Wang";
}

PersonName::PersonName(string first,
                  string last) {
        firstName = first;
        lastName = last;
}
```

https://github.com/biplav-s/course-adv-proglang/blob/main/sample-code/CandC%2B%2B/Class7and8_C%2B%2B_OO/src/implem/PersonName.cpp

Order of Calling Constructors in Hierarchy

- Parent then Child or Child before Parent ?

- **Parent first**

```
*** DEMO of Grand Child Class ***

Testing: data member –

  DEMO of Constructor – Parent Class ***

         DEMO of Constructor – Another Child Class ***

                 DEMO of Constructor – GrandChild Class ***
```

Concept: Destructors

Destructors - What is It?

- Special function in every class
  - Always has the same name as the class itself but prefixed with ~
  - Does not have an explicit return type
  - Does not take an argument
  - Maximum one destructor per class

Implementation

```
PersonName::~PersonName() {

}
```

- **Purpose**: Used to cleanup before removing objects of that class
  - Common usage: freeing memory allocated by the object’s data members before the object is destroyed
  - Common usage: Close files, streams

https://github.com/biplav-s/course-adv-proglang/blob/main/sample-code/CandC%2B%2B/Class7and8_C%2B%2B_OO/src/implem/PersonName.cpp

Order of Calling Destructors in Hierarchy

- Parent then Child or Child before Parent ?

- **Child first !**

```
*** DEMO of Grand Child Class ***

…
The grandchild's location is: AnotherChild:default-location

                 Demo of Destructor - GrandChild Class ***

         DEMO of Destructor - Another Child Class ***

  DEMO of Destructor - GrandChild Class ***
```

Full Example

Program: Class9and10_C++_OOAdv.cpp **Argument**: 3

```
*** DEMO of Grand Child Class ***

Testing: data member -

  DEMO of Constructor - Parent Class ***

         DEMO of Constructor - Another Child Class ***

                 DEMO of Constructor - GrandChild Class ***

The grandchild's name is: Parent:default-name
The grandchild's location is: AnotherChild:default-location

                 Demo of Destructor - GrandChild Class ***

         DEMO of Destructor - Another Child Class ***

  DEMO of Destructor - GrandChild Class ***
```

Discussion: Using Constructors / Destructors Effectively

- Remember: Create automatically if none provided by developer
- Constructor: initialization of data members
- Destructor: clean-up
- Remember the order, use it productively but do not overly depend on it.

Home Work 4

Due Tuesday, March 1, 2022

Home Work (#4) – C++ - Background

- Email programs parse Email headers and show content. The headers have **parts** (e.g., CC, To, From) that are part of a standard and also proprietary extensions.
- Examples for Microsoft Outlook and Gmail are shown.
- Let us assume that parts which are common to both are the standard and those unique are proprietary. So, “CC” is common and “X-MS-Has-Attach” is unique.
- Write a program, ***EmailInformationExtractor,*** which, when given a message header from either of the two programs, and a part name, will read the value of the message part.

Microsoft Outlook Header

- Received: from DS7PR19MB5853.namprd19.prod.outlook.com …
- Authentication-Results: dkim=none (message not signed
- Received: from …
- Content-Type: application/ms-tnef; name="winmail.dat"
- Content-Transfer-Encoding: binary
- From: "Sri Naga Sushmitha, Satti" <SATTI@cse.sc.edu>
- To: "Srivastava, Biplav" <BIPLAV.S@sc.edu>
- CC: "Baldwin, Randi" <baldwin@cse.sc.edu>
- Subject: Re: Possible need for … 240
- Thread-Topic: Possible need for printout for .. 240
- Thread-Index: … +AAAIRpoAAAp/ggAAAJH0=
- Date: Tue, 15 Feb 2022 13:52:33 +0000
- Message-ID: <…>
- References: …
- In-Reply-To: <…>
- Accept-Language: en-US
- Content-Language: en-US
- X-MS-Has-Attach:
- X-MS-Exchange-Organization-SCL: -1

Home Work (#4) – C++ - Requirement

- So, program name: ***EmailInformationExtractor***

- Inputs:
  - message header
  - Part name

- Output:
  - Value

- Hint
  - Use regex

Gmail Header

- Delivered-To: biplav.srivastava@gmail.com
- Received: by 2002:a05:7000:1f97:0:0:0:0 with SMTP …
- X-Google-Smtp-Source: ABdhPJz/…
- Received: from m08b.cvent-planner.com …
- From: Reply-To:To:Message-ID:Subject:MIME-Version:
- Content-Type: List-Unsubscribe; /Tvkdd8/15SWlBA=; …
- Date: Thu, 17 Feb 2022 23:56:12 +0000
- From: AAAI Staff <aaai22@aaai.org>
- Reply-To: <aaai22@aaai.org>
- To: Biplav Srivastava <biplav.srivastava@gmail.com>
- Message-ID: <..>
- Subject: AAAI-22 General Information
- MIME-Version: 1.0
- Content-Type: multipart/alternative; ..
- Content-Type: text/plain; charset=UTF-8
- Content-Transfer-Encoding: quoted-printable

Home Work (#4) – C++ - Code Design

- Create 3 classes:
  - Base class with common parts: BaseEmailHeaderType
  - Children classes with custom parts: GmailHeaderType, OutlookHeaderType
- Use exception to handle likely errors

Discussion: 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.

Core Programs Needed for Project

- Prog 1: extract data from the district **[prog1-extractor]**
- Prog 2: process it (extracted data) based on questions **[prog2processor]**
- **Prog 3: make content available in a command-line interface [prog3-ui]**
- Prog 4: handle any user query and
- Prog 5: report statistics on interaction of a session, across session

Programming Assignment # 3

- Goal: **make content available in a command-line interface** [Name: prog3-ui]

- Program should do the following:
  - Run in an infinite loop until the user wants to quit
  - Handle any user response
    - User can quit by typing “Quit” or “quit” or just “q”
    - User can enter any other text and the program has to handle it. The program should write back what the user entered and say – “I do not know this information”.
  - Handle known user query
    - “Tell me about the representative”, “Tell me about the rep” => Personal Information (Type-I2)
    - “Where does the rep live” => Contact Information (Type-I1): Home Address
    - “How do I contact my rep ” => Contact Information (Type-I1)
    - “What committees is my repo on” => Committee Assignments (Type-I3)
    - “Tell me everything” => *Give all information extracted*

Programming Assignment # 3

- Code organization
  - Create a folder in your GitHub called “prog3-ui”
  - 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 by updating Google sheet; optionally, send email to instructor and TA

- Use concepts learned in class
  - Classes
  - Exceptions
  - UML Diagrams

Example: Representative Information

Input and Output Example

prog2ui

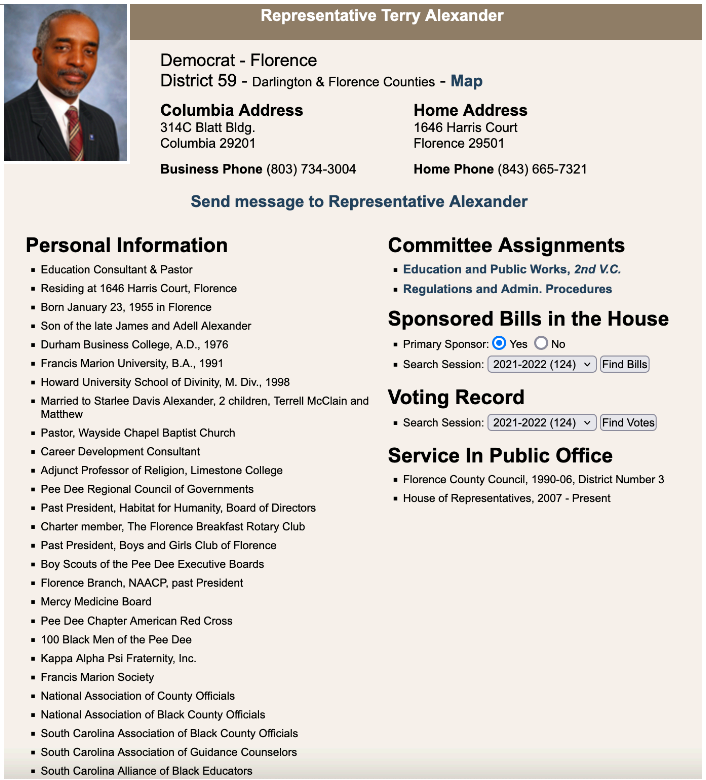
System: “Hi – Welcome”

User: “Tell me about the rep”

System: …

User: …

- Contact Information (Type-I1)
- 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)

Voting Record

- Search Session: 2021-2022 (124)

Service In Public Office

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

Concluding Section

Lecture 14: Concluding Comments

- We looked at the concept constructor
- We looked at the concept of destructor
- Home Work #4 – due Tuesday, March 1, 2022
- Programming Assignment #3 starts, due Thursday, March 3, 2022

About Next Lecture – Lecture 15

Lecture 15: Operators and Overloading

- C++ operators
- Overloading operators

14	**Feb 24 (Th)**	**OO – Constructor, Destructor**	**Prog 3 - start**
15	Mar 1 (Tu)	OO – operators, access control	HW 4 due
16	Mar 3 (Th)	C++ standard library	Prog 3 - end Semester - Midpoint