CSCE 240: Advanced Programming Techniques
Lecture 15: Operators, HW 4 (Review)

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

1ST MARCH 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 15

- Introduction Section
  - Recap of Lecture 14
  - TA and SI Updates
- Main Section
  - (Peer) Evaluation of Home work #4
  - Concept: Operators
  - Concept: Operator precedences
  - Discussion: Project – PA #3 Check
- Concluding Section
  - About next lecture – Lecture 16
  - Ask me anything

Introduction Section

Recap of Lecture 14

- We explored the concepts of
  - constructors
  - destructors
- Home Work #4 – due Tuesday, March 1, 2022
- Programming Assignment #3 starts, due Thursday, March 3, 2022

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

Home Work 4 (Peer Review)

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

Peer Review: Homework Assignment #4

1. Go to spread sheet and on ”Homework Assignments - Peer Review” tab. Go for today’s date
2. Go to the row with your name
3. Peer review (10 mins)
    1. Enter serial 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 serial 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 on HW

- Peer Code Reviewing
- Peer Testing

Concept: Operators

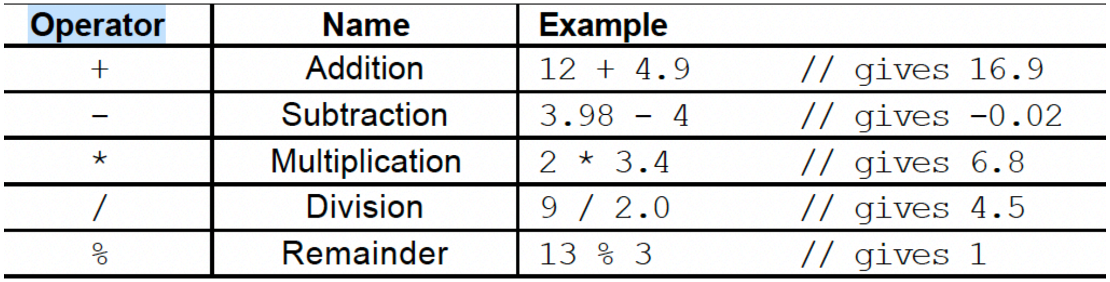
Operators - What are They?

- **Built-in special functions** to create **expressions**
- Expressions lead to computation of values
  - Can also cause side-effects
- Operators are employed over expressions called operands. Often operands are constants and variables.
- **Purpose**: create expressions in a compact and consistent, well-understood manner

Types

- Arithmetic
- Relational
- Logical
- Bitwise
- Increment/ decrement
- Assignment
- Conditional expressions
- Others

Operators - Arithmetic



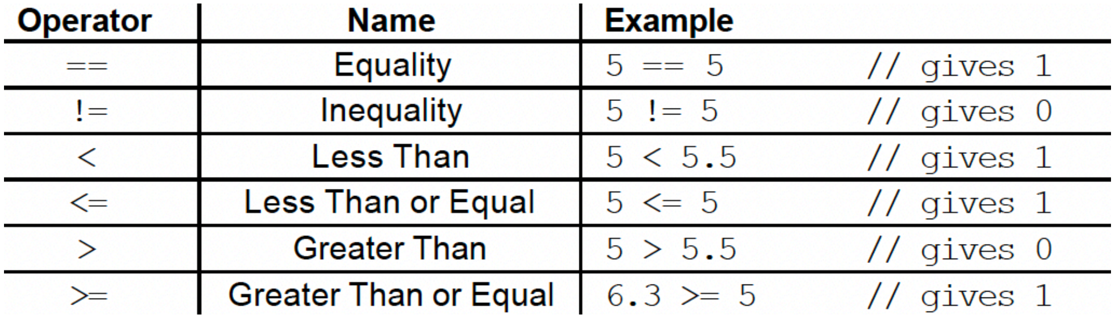
Operator	Name	Example
+	Addition	12 + 4.9 // gives 16.9
–	Subtraction	3.98 - 4 // gives -0.02
*	Multiplication	2 * 3.4 // gives 6.8
/	Division	9 / 2.0 // gives 4.5
%	Remainder	13 % 3 // gives 1

Notes

- Data type defines the nature of results – e.g., integer v/s float
- Remainder expects the two operands to be integers
- Overflow happens when outcome of an arithmetic operation to be too large for storing in a designated variable
- Division by zero error has to be handled; default is to terminate execution

Credits: C++ Essentials by Sharam Hekmat, PragSoft

Operators - Relational



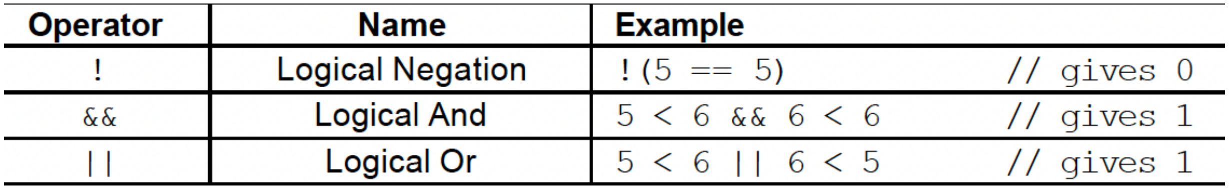
Operator	Name	Example
==	Equality	5 == 5 // gives 1
!=	Inequality	5 != 5 // gives 0
<	Less Than	5 < 5.5 // gives 1
<=	Less Than or Equal	5 <= 5 // gives 1
>	Greater Than	5 > 5.5 // gives 0
>=	Greater Than or Equal	6.3 >= 5 // gives 1

Notes

- Operands must evaluate to a number
- Comparing character works - 'A' < 'F' // gives 1 (is like 65 < 70)
- Comparing string will compare their addresses –
  ”CSCE" < ”240”, not desirable

Credits: C++ Essentials by Sharam Hekmat, PragSoft

Operators - Logical



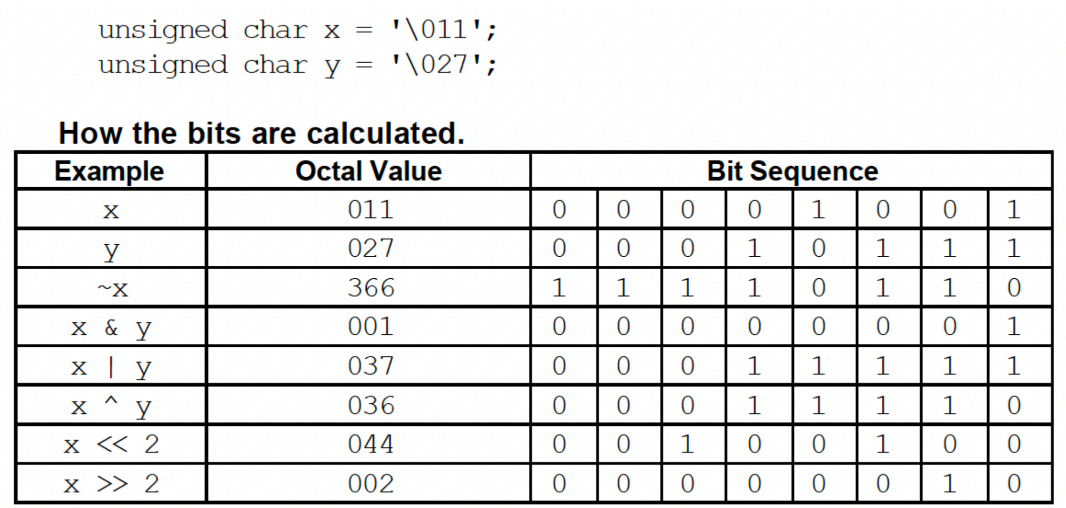
Operator	Name	Example
!	Logical Negation	`!(5 == 5)        // gives 0`
&&	Logical And	`5 < 6 && 6 < 6   // gives 1`
||	Logical Or	`5 < 6 || 6 < 5   // gives 1`

Notes

- C++ does not have boolean type
- 0 indicates a false, 1 a true

Credits: C++ Essentials by Sharam Hekmat, PragSoft

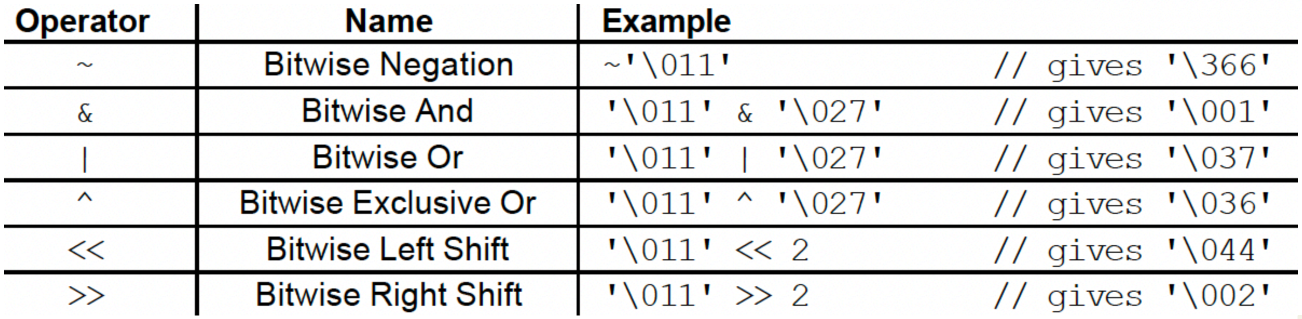
Operators - Bitwise



```
unsigned char x = '\011';
unsigned char y = '\027';
```

How the bits are calculated.

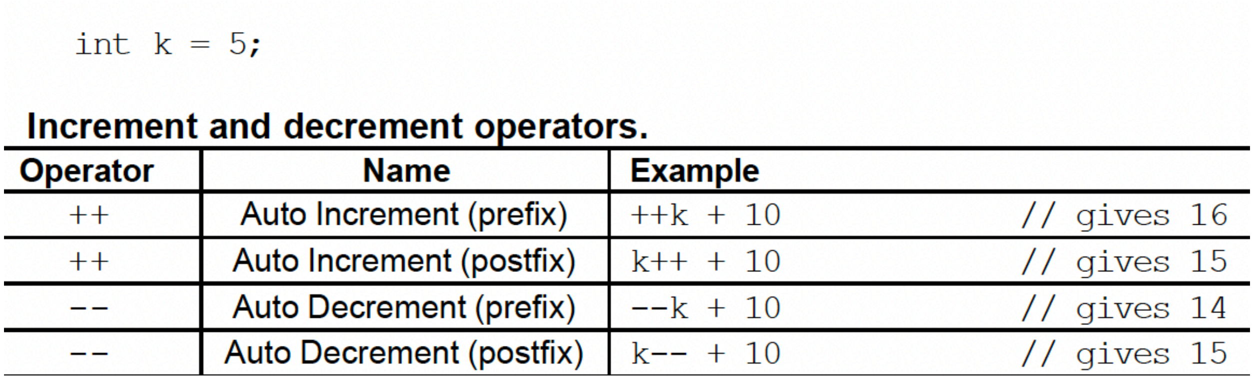
Example	Octal Value	Bit Sequence							
x	011	0	0	0	0	1	0	0	1
y	027	0	0	0	1	0	1	1	1
~x	366	1	1	1	1	0	1	1	0
x & y	001	0	0	0	0	0	0	0	1
x | y	037	0	0	0	1	1	1	1	1
x ^ y	036	0	0	0	1	1	1	1	0
x << 2	044	0	0	1	0	0	1	0	0
x >> 2	002	0	0	0	0	0	0	1	0



Operator	Name	Example
~	Bitwise Negation	`~'\011'                // gives '\366'`
&	Bitwise And	`'\011' & '\027'        // gives '\001'`
|	Bitwise Or	`'\011' | '\027'        // gives '\037'`
^	Bitwise Exclusive Or	`'\011' ^ '\027'        // gives '\036'`
<<	Bitwise Left Shift	`'\011' << 2            // gives '\044'`
>>	Bitwise Right Shift	`'\011' >> 2            // gives '\002'`

Credits: C++ Essentials by Sharam Hekmat, PragSoft

Operators – Increment/Decrement



```
int k = 5;
```

Increment and decrement operators.

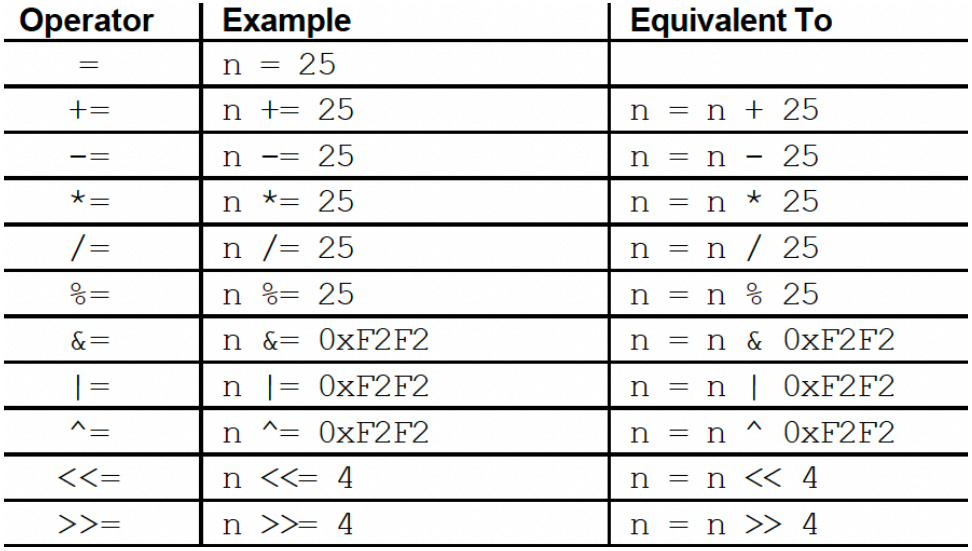
Operator	Name	Example
++	Auto Increment (prefix)	`++k + 10            // gives 16`
++	Auto Increment (postfix)	`k++ + 10            // gives 15`
--	Auto Decrement (prefix)	`--k + 10            // gives 14`
--	Auto Decrement (postfix)	`k-- + 10            // gives 15`

Notes

- Applicable to integers and real values
- Note difference between prefix and posfix

Credits: C++ Essentials by Sharam Hekmat, PragSoft

Operators – Assignment



Operator	Example	Equivalent To
=	n = 25	
+=	n += 25	n = n + 25
-=	n -= 25	n = n - 25
*=	n *= 25	n = n * 25
/=	n /= 25	n = n / 25
%=	n %= 25	n = n % 25
&=	n &= 0xF2F2	n = n & 0xF2F2
|=	n |= 0xF2F2	n = n | 0xF2F2
^=	n ^= 0xF2F2	n = n ^ 0xF2F2
<<=	n <<= 4	n = n << 4
>>=	n >>= 4	n = n >> 4

Notes

- Improves programmer productivity
- Makes code less readable
- No impact to code performance

Credits: C++ Essentials by Sharam Hekmat, PragSoft

Operators – Conditional Expressions

operand1 ? operand2 : operand3

Example

```
int m = 1, n = 2;

 int min = (m < n ? m : n);
```

Example – Same As

```
int m = 1, n = 2;

 int min;

 if  (m < n)
       min = m;
   else
       min = n
```

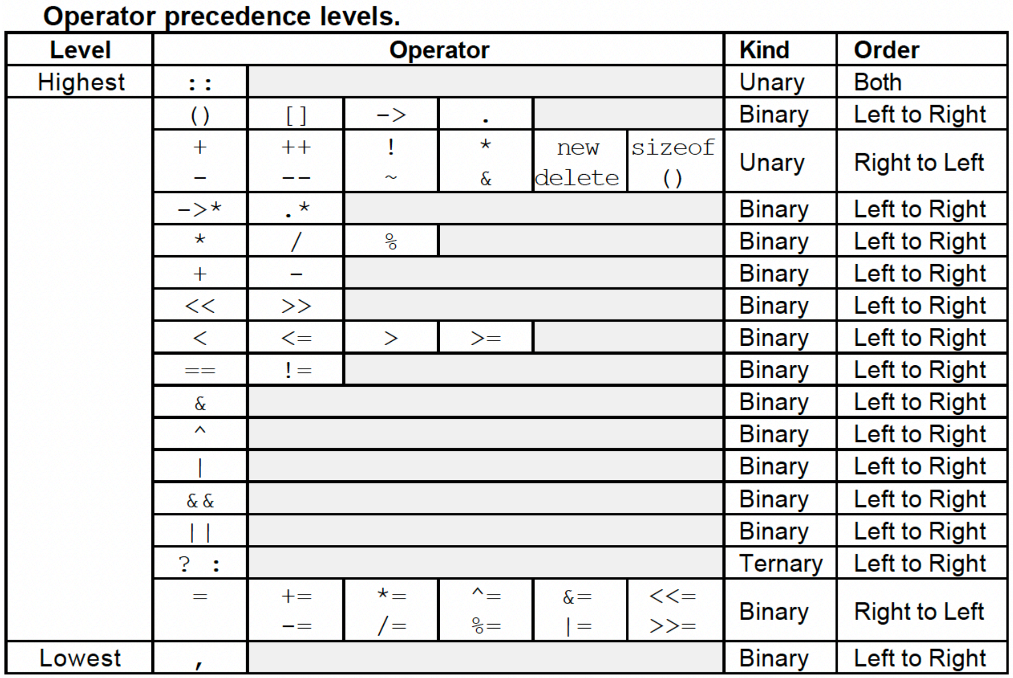
Credits: C++ Essentials by Sharam Hekmat, PragSoft

Other Operators

- Comma (,): evaluate expressions from left and then right side of comma
- sizeof(): calculating the size of any data item or type in bytes
- new(): allocate memory
- delete(): free memory

Credits: C++ Essentials by Sharam Hekmat, PragSoft

Operator Precedence



Operator precedence levels.

Level	Operator						Kind	Order
Highest	::						Unary	Both
	()	[]	->	.			Binary	Left to Right
	+ –	++ --	! ~	* &	new delete	sizeof ()	Unary	Right to Left
	->*	.*					Binary	Left to Right
	*	/	%				Binary	Left to Right
	+	–					Binary	Left to Right
	<<	>>					Binary	Left to Right
	<	<=	>	>=			Binary	Left to Right
	==	!=					Binary	Left to Right
	&						Binary	Left to Right
	^						Binary	Left to Right
	|						Binary	Left to Right
	&&						Binary	Left to Right
	||						Binary	Left to Right
	? :						Ternary	Left to Right
	=	+= -=	*= /=	^= %=	&= |=	<<= >>=	Binary	Right to Left
Lowest	,						Binary	Left to Right

Expression

a == b + c * d

Same as

a == (b + (c * d))

Credits: C++ Essentials by Sharam Hekmat, PragSoft

What More ?

- Recall <<
  - Example: cout << “Hello World! << endl;

- Operator overloading
  - To be covered in future lecture

Discussion: Course Project

PA #3 Check: Due Thursday, March 3, 2022

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

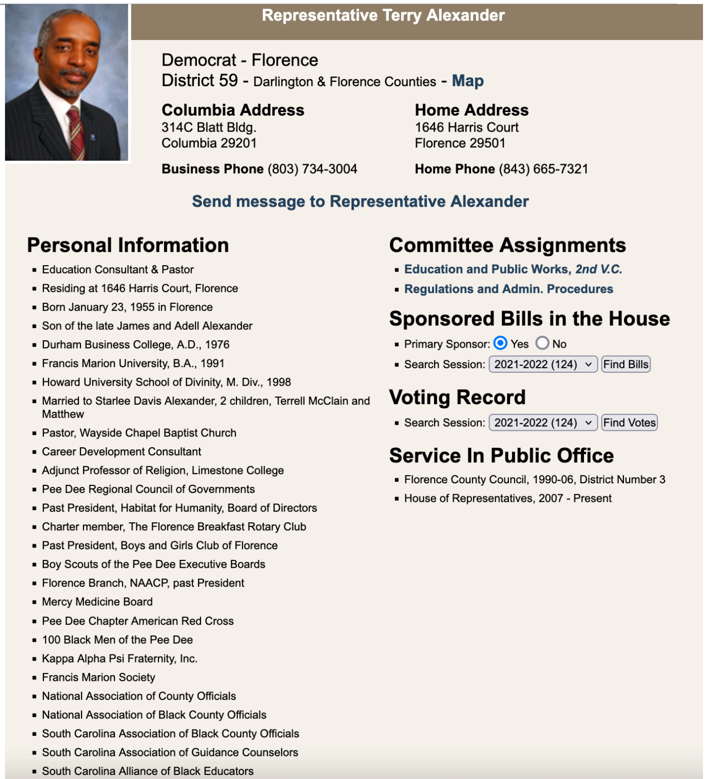
System: “Hi – Welcome”

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) ⌄ 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

Concluding Section

Lecture 15: Concluding Comments

- Reviewed HW#4
- We looked at the concept of operators
  - Many types
  - Precedence order when evaluating
- Reminder: Programming Assignment #3 due Thursday, March 3, 2022

About Next Lecture – Lecture 16

Lecture 16: C++ Standard Libraries

- C++ standard library

Mar 3 (Th)	**C++ standard library**	**Prog 3 - end Semester - Midpoint**
Mar 8 (Tu)		Spring break – No class
Mar 10 (Th)		Spring break – No class
Mar 15 (Tu)	Testing strategies	Prog 4 - start
Mar 17 (Th)	Advanced: Pointers	HW 5 due