

CSCE 240: Advanced Programming Techniques

Lecture 15: Operators, Review of Quiz/ PAs, PA3 Due

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

29TH FEBRUARY 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 16

- Introduction Section
 - Recap of past 3 Lectures
- Main Section
 - Concept: Operators
 - Concept: Operator precedences
 - Discussion: Project – PA #3 Check
- Concluding Section
 - About next lecture – Lecture 17
 - Ask me anything

Introduction Section

AAAI 2024 and AI Research on Campus

- Overall: <https://aaai.org/aaai-conference/>
 - Under-Graduates program: <https://aaai.org/aaai-conference/undergraduate-consortium-program/>
- AI4Society at AAAI 2024 (<https://ai4society.github.io/>) :
 - Tutorial (LLMs for planning), a workshop (AI and elections), a deployed application paper (ULTRA), and a demonstration paper (AI planning for information spread in social networks).
 - Awards
 - Expressive and Flexible Simulation of Information Spread Strategies in Social Networks Using Planning – AAAI Best Demo Award at AAAI. It enables detailed simulations of opinion evolution and strategic interventions using planning. For further details:
 Video: <https://lnkd.in/gV-WhpDU>,  Poster: https://lnkd.in/gAkxDG_5,  Paper: <https://lnkd.in/e7bFjHdP>
 - Deployed application award for “Promoting Research Collaboration with Open Data Driven Team Recommendation in Response to Call for Proposals”. Paper: <https://lnkd.in/gEj2jXBx>
 - Looking for 1-2 undergrads to work in summer on AI algorithmic issues; send note/ talk to instructor; funding - on-campus funding programs (McNair/ Magellan/ ...) + top-up by instructor

Recap of Lectures 13-15

- L13
 - Review for quiz 1
 - Start of PA3
- L14
 - Quiz 1
- L15
 - Project reviews, presented by students
- Programming Assignment #3 ends, due today

Review Quiz 1

Main Section

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

Code Sample

[https://github.com/biplav-s/course-adv-proglang/blob/main/sample-code/CandC%2B%2B/Class15and16 OperatorSTL/src/Class15and16 OperatorSTL.cpp](https://github.com/biplav-s/course-adv-proglang/blob/main/sample-code/CandC%2B%2B/Class15and16%20OperatorSTL/src/Class15and16%20OperatorSTL.cpp)

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 postfix

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	sizeof	Unary	Right to Left	
	-	--	~	&	delete	()			
	->*	.*					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

Exercise: Creating Own Operator

// An object of this type represents a linear function of one variable $a * x + b$.

```
struct Linear
{
    double a, b;
    double operator()(double x) const {
        return a * x + b;
    }
};
```

Credit: <https://en.cppreference.com/w/cpp/language/operators>

What More ?

- Recall <<
 - Example: `cout << "Hello World! << endl;`
- Operator overloading
 - To be covered in future lecture

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.

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 append, saying – “– I do not know this information”.
 - Handle known user query types
 - “Tell me about **IBM**” or “What are the risk factor for **IBM**?” => (Part 1), or (Part 1: Item 2), accordingly
 - “What markets does **IBM** operate in?”, “Are there any disclosures from **IBM**?” => (Part 2)
 - “who are the directors? ” => (Part 3: Item ..) // assume company, or tell of all companies, or ask ...
 - “Tell me about **IBM’s** statements” => (Part 4)
 - ...
 - “**Tell me everything**” => *Give all information extracted*

Concepts: 10-K, Parts, Items

Parts

- Part 1: Business Background and Risks
 - Item 1: Business
 - Item 2: Risk factors
 - **Item 3: Properties**
 - **Item 4: Legal Proceedings**
- Part 2: Operations and Disclosures
 - .. Market
 - .. Disclosures
- Part 3: Company Structure
 - Directors
 - Compensation
- Part 4: Financial Statements
 - Statements

Notes on PA#3

- Handle all parts and items
 - Multiple ways to ask for same information type
 - Variant assumes company's name from context or is specified
- Handle special query: *Tell me everything*
- Handle others
 - Chit-chat
 - Give controlled response under all condition

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

Content Reference: Queries for (Answers) Data We Have

- What does the (company) do? // Answers in Part 1
 - What is the (company's) business?
 - What are (company's) risk factors?
 - What does (company) own?
 - ...
- Where does (company) operate? // Answers in Part 2
 - What has (company) disclosed?
- How is (company) structured? // Answers in Part 3
 - Who is (company's) CEO?
 - How much does (person) earn?
 - ...
- What was in (company) statements? // Answers in Part 4
 - ...

Concepts: 10-K, Parts, Items

Parts

- Part 1: Business Background and Risks
 - Item 1: Business
 - Item 2: Risk factors
 - Item 3: Properties
 - Item 4: Legal Proceedings
- Part 2: Operations and Disclosures
 - .. Market
 - .. Disclosures
- Part 3: Company Structure
 - Directors
 - Compensation
- Part 4: Financial Statements
 - Statements

Concluding Section

Lecture 16: Concluding Comments

- Reviewed Quiz 1
- We looked at the concept of operators
 - Many types
 - Precedence order when evaluating
- Reminder: Programming Assignment #3 due today by 10pm

About Next Lecture – Lecture 17

Lecture 17: C++ Standard Libraries

- C++ standard library

12	Feb 15 (Th)	OO – Constructor, Destructor	Prog 2 – end
13	Feb 20 (Tu)	Review: inheritance, Polymorphism	Prog 3 - start
14	Feb 22 (Th)	In class test	Quiz 1 – In class
15	Feb 27 (Tu)	In class Project Review: PA1 and PA2	
16	Feb 29 (Th)	OO – operators, access control	Prog 3 - end Semester - Midpoint
	Mar 5 (Tu)		Spring break – No class
	Mar 7 (Th)		Spring break – No class
17	Mar 12 (Tu)	C++ standard library, Testing strategies	Prog 4 - start
18	Mar 14 (Th)	Advanced: Pointers	HW 4 due
19	Mar 19 (Tu)	Advanced: Pointers, I/O	
20	Mar 21 (Th)	Advanced: Operator overloading	Prog 4 – end (March 26, 2023)