

CSCE 240: Advanced Programming Techniques

Lecture 10: Object Oriented Concepts – Polymorphism, Regular Expressions, HW 3 (Review)

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

8TH 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 10

- Introduction Section
 - Recap of Lecture 9
- Main Section
 - Concept: Regular expressions
 - Discussion: Project discussion
 - Concept: Polymorphism
- Concluding Section
 - About next lecture – Lecture 11
 - Ask me anything

Introduction Section

Recap of Lecture 9

- We relooked at the concept of inheritance
- We discussed home work assignment #3 – due next Thursday (Feb 16)
- We discussed programming assignment (PA) #2

About Programming Languages in Course

- C++ is the main language for the course.
 - Used to demonstrate concepts and expect everyone to know it at the level that they can do peer evaluation and testing of each other's code in **home assignments**.
 - **For projects and programming assignments**, students have option to code in Java or Python as well.
- Cross-language understanding of concepts
 - Code in multiple languages is sometimes shown to demonstrate generality of concepts and specific peculiarities in implementation
 - UML diagrams will be used to convey cross-language concepts as well

In quizzes,

- **Questions will be about concepts, pseudo-code and UML diagram.**
- C++ code fragments may be shown or asked to be written, but they do not have to be running code. The quizzes will be in class and can be done on paper or a text editor like Google doc.

Main Section

Concept: Regular Expressions

Example #1

- **String:** "Advanced Programming Techniques"
 - Is "Prog" in this string ? => string match
 - What are the characters after "Prog"? => string match followed by sub-string extraction
- **Output - Yes**
 - string – 'Advanced Programming Techniques' matches pattern - **(.*)(Prog)(.*)**
- **Solver the problem by**
 - Is "Prog" in this string ? => string match
 - What are the characters after "Prog"? => string match followed by sub-string extraction

Example #2

- String 1: "AdvProg.pdf"
 - Is this a valid file name?
- String 2: " Adv Prog.pdf "
 - Is this a valid file name?
- Pattern = "[a-zA-Z_][a-zA-Z_0-9]*\\. [a-zA-Z0-9]+" ;
- Output – Yes (String 1) and No (String 2)

Review: Regular Expression

Metacharacter	Explanation
^	Matches the starting position within the string
.	Matches any single character
[]	Matches a single character that is contained within the brackets
[^]	Matches a single character that is not contained within the brackets.
\$	Matches the ending position of the string
*	Matches the preceding element zero or more times
+	Matches the preceding element one or more times
	Separates choices

Regex	Matches any string that
hello	contains {hello}
gray grey	contains {gray, grey}
gr(a e)y	contains {gray, grey}
gr[ae]y	contains {gray, grey}
b[aeiou]bble	contains {babble, bebble, bibble, bobble, bubble}
[b-chm-pP]at ot	contains {bat, cat, hat, mat, nat, oat, pat, Pat, ot}
colour?r	contains {color, colour}
rege(x(es)? xps?)	contains {regex, regexes, regexp, regexps}
go*gle	contains {ggle, gogle, google, gooogle, goooogle, ...}
go+gle	contains {gogle, google, gooogle, goooogle, ...}
g(oog)+le	contains {google, googoogle, googoogle, googoogle, ...}
z{3}	contains {zzz}
z{3,6}	contains {zzz, zzzz, zzzzz, zzzzzz}
z{3,}	contains {zzz, zzzz, zzzzz, ...}

Example Source: <https://cs.lmu.edu/~ray/notes/regex/>

Regular Expression in C++

- Part of stdlib

References:

<https://www.softwaretestinghelp.com/regex-in-cpp/>

<https://www.cplusplus.com/reference/regex/>

Implementation: Finding Words in Python

- Python has extended Regex specifications for convenience
- Useful for
 - Matching patterns
 - Information extraction
 - Content manipulation (e.g., substitution)
 - Error (e.g., spelling) correction

```
data = "The CSCE 771 course is taught at  
University this Fall!"  
pattern = "[tT]+\w"  
m = re.findall(pattern, data)  
print(m)
```

```
['Th', 'ta', 'ty', 'th']
```

Details: <https://docs.python.org/3/library/re.html>

Regex Python Code Examples

- See
- More regular expression examples
 - <https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l20-text-overview/WordLesson-Examples.ipynb>

Regular Expression in Java

- Part of java.util.regex library
- (Matcher, Pattern) are the most important classes
- **Note:** same regex expression (pattern) works across languages

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
- **Prog 2: process it (extracted data) 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

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

Programming Assignment # 2

- Goal: **process extracted text based on questions**
 - Language of choice: Any from the three (C++, Java, Python)
- Program should do the following:
 - Take input from a local file which has content obtained from Prog#1 (when **company** name given as input)
 - Given an information type as input, the program will return its content
 - Examples: what is company's risk factors? What does company's CEO earn?
 - Input type can be given as command line argument.
Examples:
 - `prog2processor -t "what are IBM's risk factors?"` // Tell about company
 - `prog2processor -t "all information"` // Get all info for a company
- For demonstrating that your program works, have a file called "test_output.txt" showing the set of supported commandline options and output in the doc folder.

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

• Code organization

- Create a folder in your GitHub called "prog2-processor"
- 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

Hint: Use Regex for Information Extraction

- Use regex to find information types and sub-types
- Make a set of patterns (regex) for information of interest
 - Think of patterns as “rules” .e.g., phone number comes after “Phone”
 - Patterns for information types and sub-types
 - Ordering of rules may matter if multiple rules match
- If pattern is found, extract content of interest near it using string operation

Concept: Polymorphism

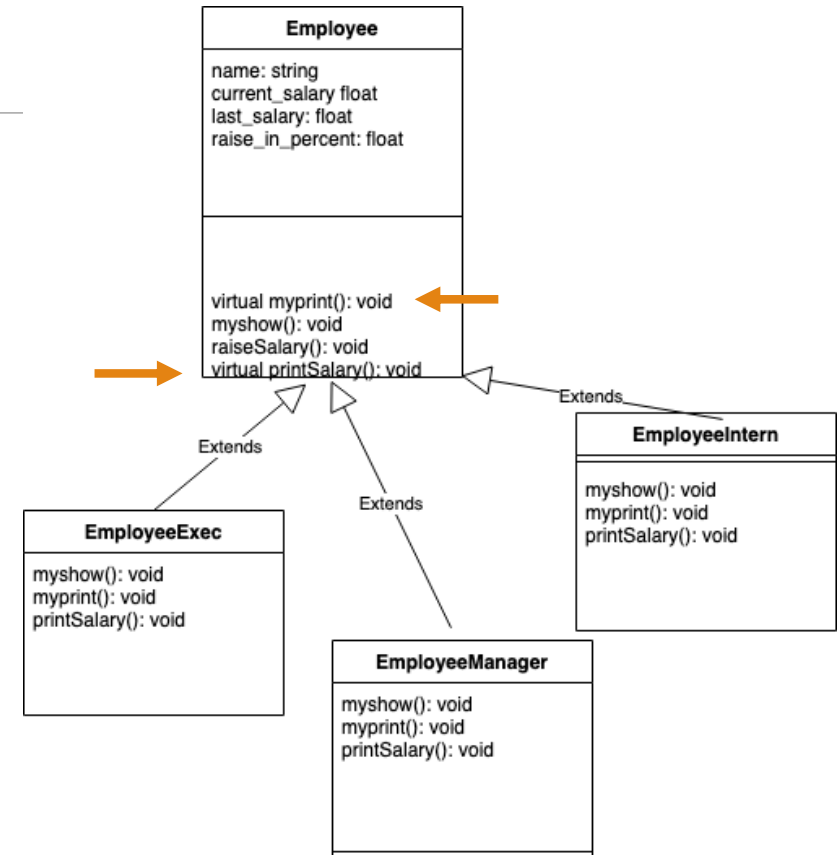
“Multiple shapes”

What is Polymorphism ?

- A class “inheriting” or reusing **characteristics** from another, existing class, dynamically depending on how the method is declared !
- In contrast, inheritance discussed until now was static

Why Use Polymorphism ?

- Promote reuse
- Make code understandable, improve maintainability
- Promote security and data integrity
- Improve testing
- Improve code development productivity
- **Context-dependent customization of inheritance**



Credits: Based on code at

- <https://www.geeksforgeeks.org/polymorphism-in-c/>
- <https://www.geeksforgeeks.org/virtual-functions-and-runtime-polymorphism-in-c-set-1-introduction/>

How to Use Polymorphism ?

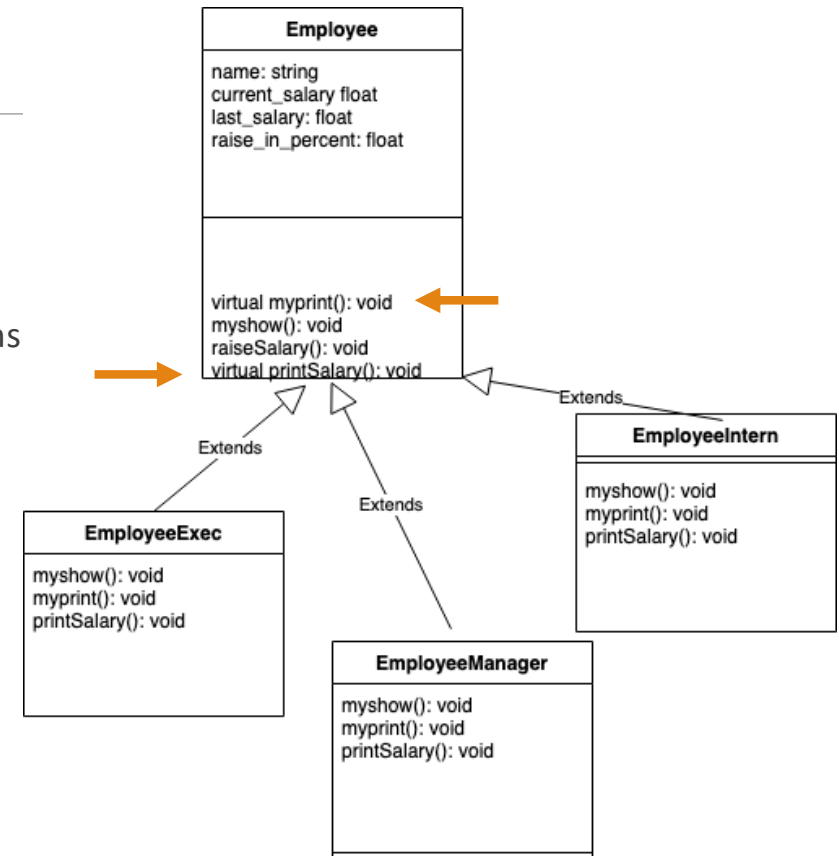
- Language independent syntax
- Illustration
 - 4 classes; 1 base, 3 derived
 - Basic: no data members; myshow() and myprint() functions
 - Advanced: 3 data members, printSalary() function

```
Employee: myprint base class
Employee: myshow base class

EmployeeManager: myprint derived class
Employee: myshow base class

EmployeeIntern: myprint derived class
Employee: myshow base class

EmployeeExec: myprint derived class
Employee: myshow base class
```



Key Points - Polymorphism

1. The method must appear in a class that is part of an inheritance hierarchy
2. The method must declared virtual in the base class at the top of the hierarchy
3. Derived classes override the behavior of the inherited virtual methods as needed.
4. Clients must invoke the method via a pointer (or reference) to an object, not directly through the object itself

Code:

<https://github.com/biplav-s/course-adv-proglang/tree/main/sample-code/CandC%2B%2B/Class9and10%20OOAdv/src>

Credit: Fundamentals of Programming C++, Richard L. Halterman

Notes on Polymorphism

- Support for Polymorphism is not uniform across languages
- C++ is most expressive; controlled by virtual; allows dynamic binding (change of behavior)
- Java and Python have limited support; does static binding

Concluding Section

Lecture 10: Concluding Comments

- Understood concept of regular expressions
- Saw how regex could be helpful for Prog. Assignment #2 (due Feb 15)
- Looked at the concept of Polymorphism

About Next Lecture – Lecture 11

Lecture 11: Exceptions

- Concept of errors, exceptions
- Controlling program behavior when it may give errors/ exceptions

Feb 1 (Th)	Code org (C++)	Prog 1 - end
Feb 6 (Tu)	OO – inheritance	Prog 2 - start
Feb 8 (Th)	Regex, OO - polymorphism	HW 3 due
Feb 13 (Tu)	Exceptions	
Feb 15 (Th)	Review: inheritance, Polymorphism	Prog 2 – end