

# CSCE 240: Advanced Programming Techniques

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

---

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

9<sup>TH</sup> 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 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 rellooked 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 – 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.



# 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

# 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 with whose content is obtained from Prog#1 (when **disease** name given as input)
  - Given an information type as input, the program will return its content
    - Examples: what is disease (I1), who is at risk (I2), disease vaccine (I12)
    - Input type can be given as command line argument.  
Examples:
      - prog2processor -t "what is **malaria**?" // Tell about disease
      - prog2processor -t "more information" // Get more info
  - 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.
- 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

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

- What is malaria? [I1]
- Who is at risk? [I2]
- What can travelers do to prevent malaria? [I3]
- After Travel [I4]
- More Information [I5]

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

- [What Is Malaria?](#) [I1]
- [Malaria Causes and Risk Factors](#) [I2]
- [Types of Malaria](#) [I6]
- [Symptoms](#) [I7]
- [When to Call a Doctor About Malaria](#) [I8]
- [Malaria Diagnosis](#) [I9]
- [Malaria Treatment](#) [I10]
- [Malaria Complications](#) [I11]
- [Malaria Vaccine](#) [I12]

# Example: Representative Information

## Input:

```
prog2processor -t "what is malaria?" // Tell about disease
```

## Output:

Malaria is a serious and sometimes fatal disease caused by a parasite that commonly infects a certain type of mosquito which feeds on humans. People who get malaria are typically very sick with high fevers, shaking chills, and flu-like illness. Four kinds of malaria parasites infect humans: *Plasmodium falciparum*, *P. vivax*, *P. ovale*, and *P. malariae*. In addition, *P. knowlesi*, a type of malaria that naturally infects macaques in Southeast Asia, also infects humans, causing malaria that is transmitted from animal to human ("zoonotic" malaria). *P. falciparum* is the type of malaria that is most likely to result in severe infections and if not promptly treated, may lead to death. Although malaria can be a deadly disease, illness and death from malaria can usually be prevented.

About 2,000 cases of malaria are diagnosed in the United States each year. The vast majority of cases in the United States are in travelers and immigrants returning from parts of the world where malaria transmission occurs, including sub-Saharan Africa and South Asia.

...

<https://www.cdc.gov/malaria/about/faqs.html>

### The Disease

#### What is malaria?

Malaria is a serious and sometimes fatal disease caused by a parasite that commonly infects a certain type of mosquito which feeds on humans. People who get malaria are typically very sick with high fevers, shaking chills, and flu-like illness. Four kinds of malaria parasites infect humans: *Plasmodium falciparum*, *P. vivax*, *P. ovale*, and *P. malariae*. In addition, *P. knowlesi*, a type of malaria that naturally infects macaques in Southeast Asia, also infects humans, causing malaria that is transmitted from animal to human ("zoonotic" malaria). *P. falciparum* is the type of malaria that is most likely to result in severe infections and if not promptly treated, may lead to death. Although malaria can be a deadly disease, illness and death from malaria can usually be prevented.

About 2,000 cases of malaria are diagnosed in the United States each year. The vast majority of cases in the United States are in travelers and immigrants returning from parts of the world where malaria transmission occurs, including sub-Saharan Africa and South Asia.

Globally, the World Health Organization estimates that in 2020, 241 million clinical cases of malaria occurred, and 627,000 people died of malaria, most of them children in Africa. Because malaria causes so much illness and death, the disease is a great drain on many national economies. Since many countries with malaria are already among the poorer nations, the disease maintains a vicious cycle of disease and poverty.

[Top of Page](#)

# 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”

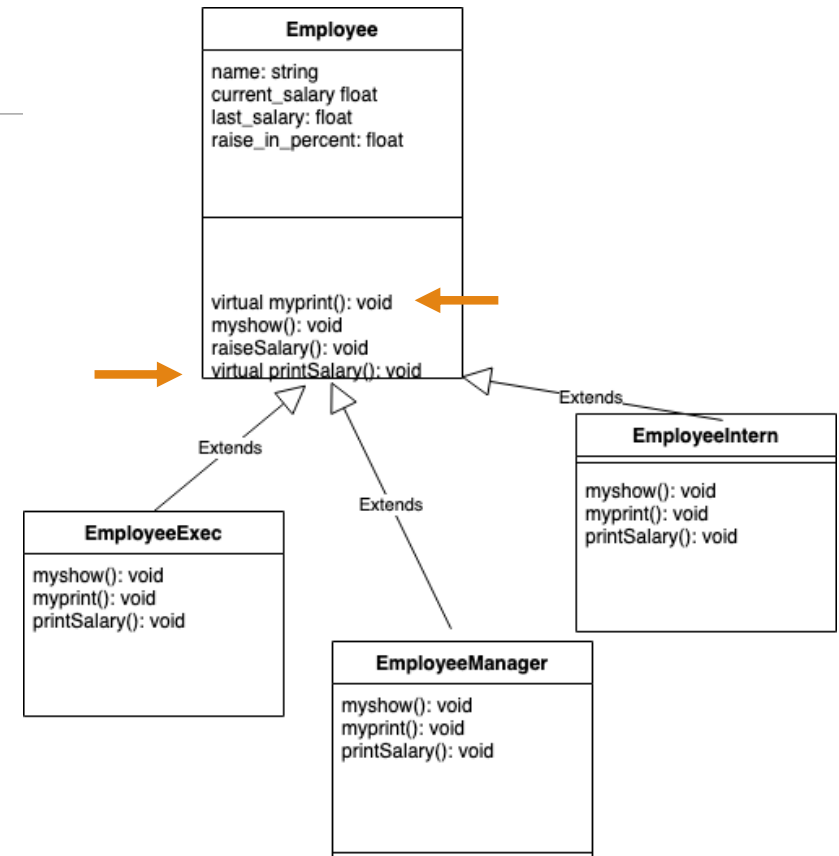
# What is Polymorphism ?

---

- 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 ?

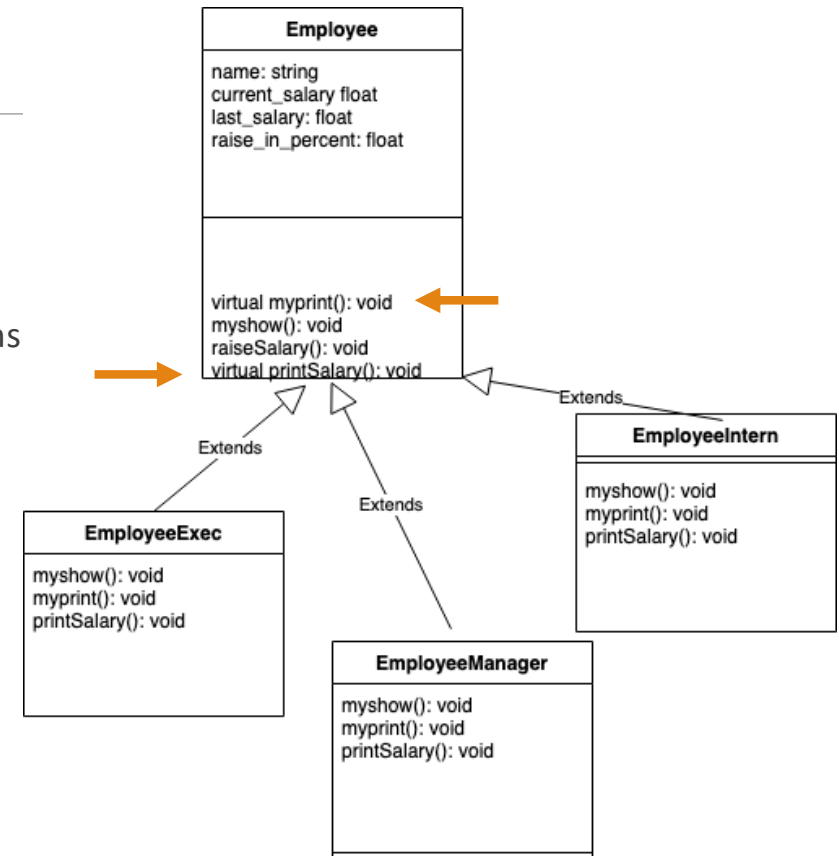
- Language independent syntax
- Illustration
  - 4 classes; 1 base, 3 derived
  - Basic: no data members; myshow() and myprint() functions
  - Advanced: 3 data members, printSalary() function

```
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 C%2B%2B OOADv/src](https://github.com/biplav-s/course-adv-proglang/tree/main/sample-code/CandC%2B%2B/Class9and10%20C%2B%2B%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 23)
- Looked at the concept of Polymorphism

# About Next Lecture – Lecture 11

---

# Lecture 11: Quiz 1 (remote)

---

- **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.

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