CSCE 240: Advanced Programming Techniques
Lecture 12: Review Object Oriented Concepts – Inheritance, Polymorphism

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

16TH 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 12

- Introduction Section
  - Recap of Lecture 11 – Quiz 1
  - AAAI 2023 – Undergraduate Consortium
- Main Section
  - Code peer review and testing: Home work #3
  - Review: Inheritance
  - Review: Polymorphism
  - Review: Regex
- Concluding Section
  - About next lecture – Lecture 13
  - Ask me anything

Introduction Section

Recap of Lecture 11

- Quiz 1
    - Feedback and discussion

AAAI 2023 – Highlights

- Overall: https://aaai-23.aaai.org/
  - Under-Graduates: https://aaai-23.aaai.org/undergraduate-consortium/

Main Section

Home Work 3

Home Work (#3) – C++

Home Work #2

- Write a program called GeometricPropertyCalculator.
  - The program reads an input file (called input.txt). Each line in the file contains dimensions of a geometric shape – rectangle, shape and triangle. Specifically:
    - For rectangle, it contains – RECTANGLE <length-in-cm> <breadth-in-cm>
    - For circle, it contains – CIRCLE <radius-in-cm>
    - For triangle, it contains – TRIANGLE <side-1-in-cm> <side-2-in-cm> <side-3-in-cm>
  - The user specifies the property to calculate as argument to the program: 1 for AREA and 2 for PERIMETER
  - The program writes output lines to an output file (called output.txt) for each shape that it reads and the property – AREA or PERIMETER.
    - For example, for RECTANGLE and property as AREA, the program should write – RECTANGLE AREA <calculated value>
  - Write GeometricPropertyCalculator in C++
    - It should support RECTANGLE, CIRCLE and TRIANGLE
    - It should support properties AREA and PERIMETER
    - If there is insufficient information, the program should give an error. E.g. TRIANGLE AREA "Not enough information to calculate"

Home Work #3

- Build a program called **OOGeometricPropertyCalculator**
  - Your new code will do the same as Home Work#2 but with OO design
  - It will have 4 classes: Shape – the parent, and its three children - Rectangle, Circle and Traingle
  - Shape will have three members: **area**, **perimeter** and **errorMessage**; and at least three functions getArea(), getPerimeter() and getErrorMessage().
  - In your code, there will be a utility file (OOGeometricPropertyCalculator.cpp) with main() and will call the classes and functions. You can choose to have one or more files for the classes.
    (E.g, For the 4 classes, 4 headers + 4 .cpp files).
  - You will also draw UML class diagrams for it
- Functionality Reminder
  - The user specifies the property to calculate as argument to the program: 1 for AREA and 2 for PERIMETER
  - The program writes output lines to an output file (called output.txt) for each shape that it reads and the property – AREA or PERIMETER.

Home Work (#3) – C++

- Code guidelines for the OO code you will write
  - Have sub-directories in your folder
    - src sub-folder, (or code) for code
    - data sub-folder, for input.txt and output.txt
    - doc sub-folder, for documentation on what the code does or sample output.

- In documentation
  - **Have a UML class diagram for the classes**
  - Observe how long was the code earlier and now. If you have to add a new functionality (like getVertices() to get all the vertices in a shape), how easy or hard will it be in HW2 code or HW3 code?

Peer Review: Homework Assignment #3

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

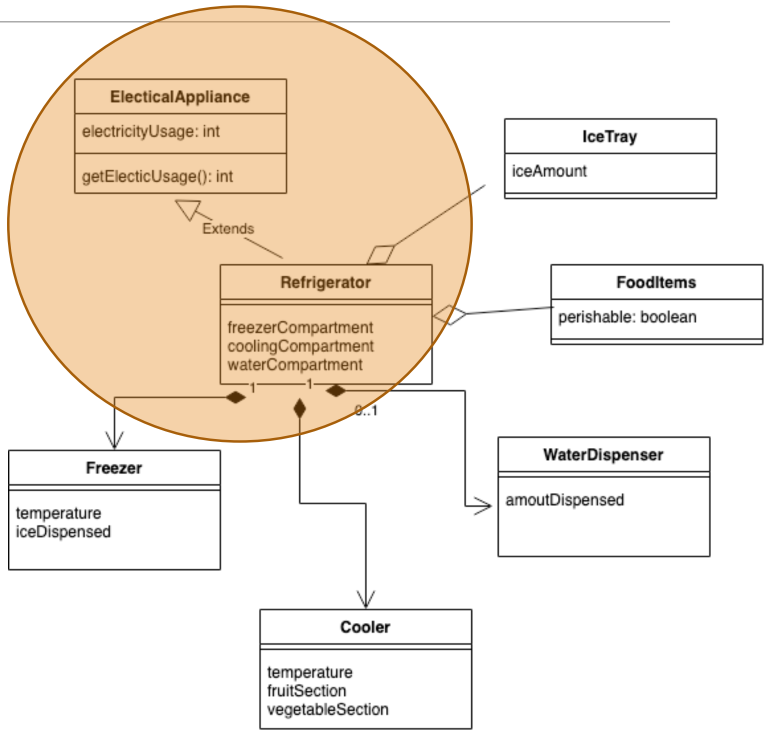
Review of Concept: Inheritance

What is Inheritance ?

- A class “inheriting” or reusing **characteristics** from another, existing class
- Synonyms: subclassing, specialization, derived
- Analogy: child inheriting from a parent
  - “Course-CSCE-240” sub-class of “Course-Undergraduate”
  - “USA” specialization of “Country”
- What are characteristics
  - Data members
    - Enrollment, timing, syllabus: course domain
    - Capital, head-of-state, currency: country domain
  - Functions manipulating the data members

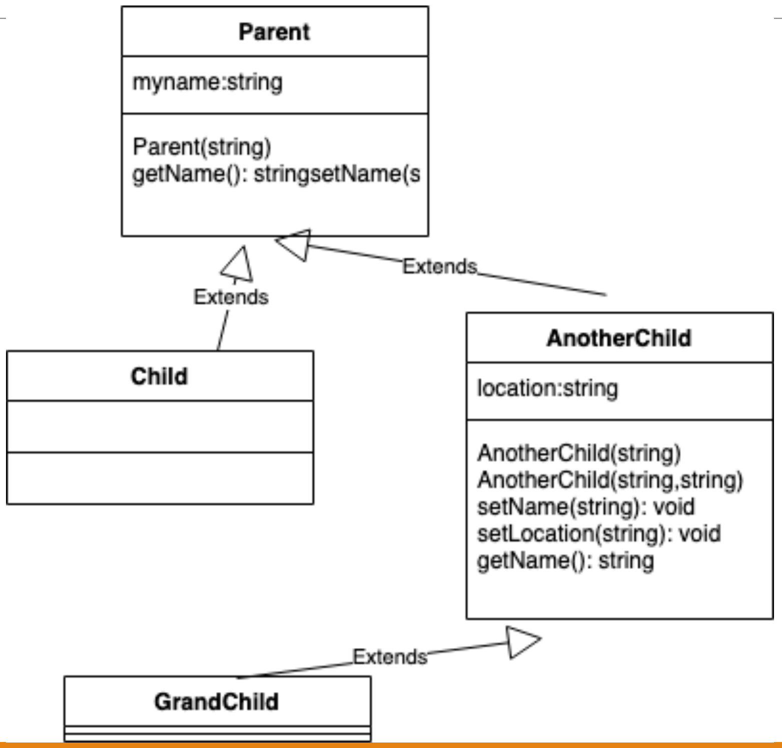
Why Use Inheritance ?

- Promote reuse
- Make code understandable, improve maintainability
- Promote security and data integrity
- Improve testing
- Improve code development productivity



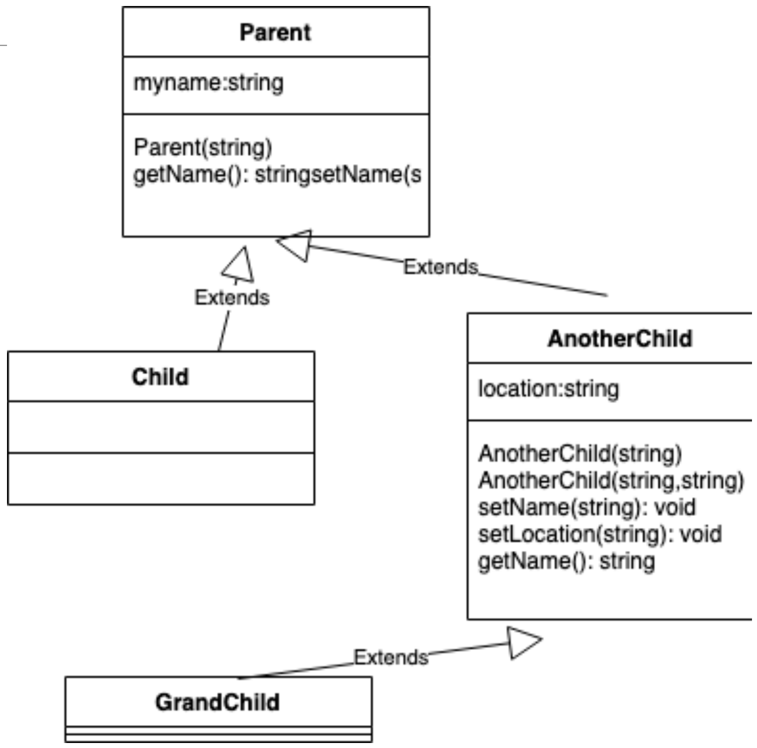
How to Use Inheritance ?

- Language independent syntax
- Illustration
  - 4 classes
  - 2 data members: myname, location
  - Access restrictions: private, protected, public



Notes on Inheritance

- Code for classes Child and GrandChild are minimal
  - Code reuse happens by default
- A child can override the behavior of its parent



Inheritance Type

- The access control levels (public, protected and private) in a class can be modified by inheritance types.
- Three inheritance types: public, protected, private
  - In public, all methods and members inherited from the parent maintain their access control level
  - In protected, all methods and members inherited from the parent maintain protected or lower access control level
  - In private, all methods and members inherited from the parent maintain private access control level
- **By default, we had been working with public inheritance types**

Access \ Inheritance Type	**public**	**protected**	**private**
public	public	protected	private
protected	protected	protected	private
private	private	private	private

Running Example - Name

- Scope: class diagram to cover names for humans and robots
  - Members: First, Middle, Last, Prefix, Suffix, Nickname, PreferredName
  - Methods: Getters and Setters
- Classes: AllNames, HumanNames, RobotNames
- Possible design
  - AllName: (PreferredName)
  - HumanNames (First, Middle, Last, Prefix, Suffix, Nickname)
  - RobotNames (PreferredName)
- Exercise:
  - Think of access and inheritance types

Review of Concept: Polymorphism

“Multiple shapes”

What is Polymorphism ?

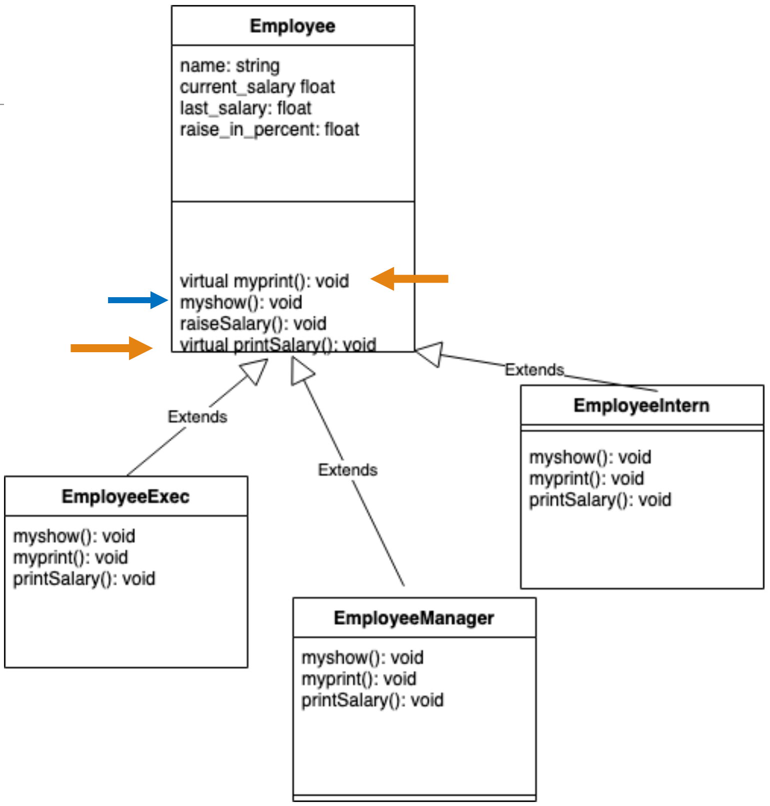
- A class “inheriting” or reusing **characteristics** from another, existing class, dynamically depending on how the method is declared !
- In contrast, inheritance discussed until now was static

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**

Code: https://github.com/biplav-s/course-adv-proglang/tree/main/sample-code/CandC%2B%2B/Class9and10_C%2B%2B_OOAdv/src

```
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 ?

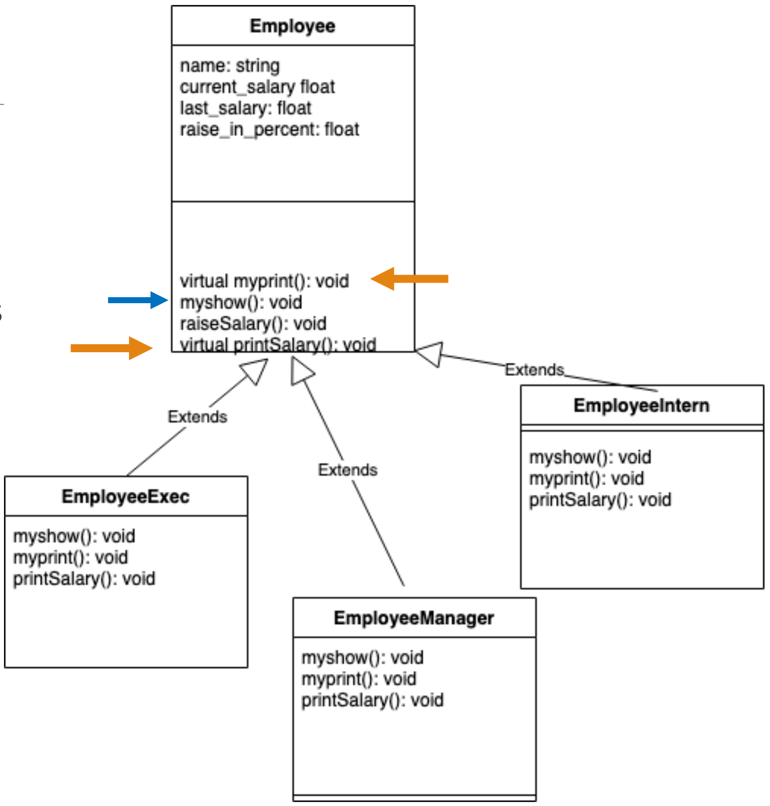
- Language independent syntax
- Illustration
  - 4 classes; 1 base, 3 derived
  - Basic: no data members; myshow() and myprint() functions
  - Advanced: 3 data members, printSalary() function

```
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

```
for (int i = 0; i < 4; i++) {
// Polymorphic Call: Calls myprint()
// according to the actual object, not
// according to the type of pointer
emps[i]->myprint();


// Polymorphic Call: Calls myshow()
// according to the actual object, not
// according to the type of pointer
emps[i]->myshow();
}
```

```
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
```

Code: https://github.com/biplav-s/course-adv-proglang/tree/main/sample-code/CandC%2B%2B/Class9and10_C%2B%2B_OOAdv/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

```
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
```

Review of Concept: Regex

“Regular Expressions”

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}
colou?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, googoogoogle, googoogoogoogle, ...}
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/

Regex - Code Demo

Code: https://github.com/biplav-s/course-adv-proglang/blob/main/sample-code/CandC%2B%2B/Class9and10_C%2B%2B_OOAdv/src/Class9and10_C%2B%2B_OOAdv.cpp

Argument: [^0-3]

What does this mean?

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:

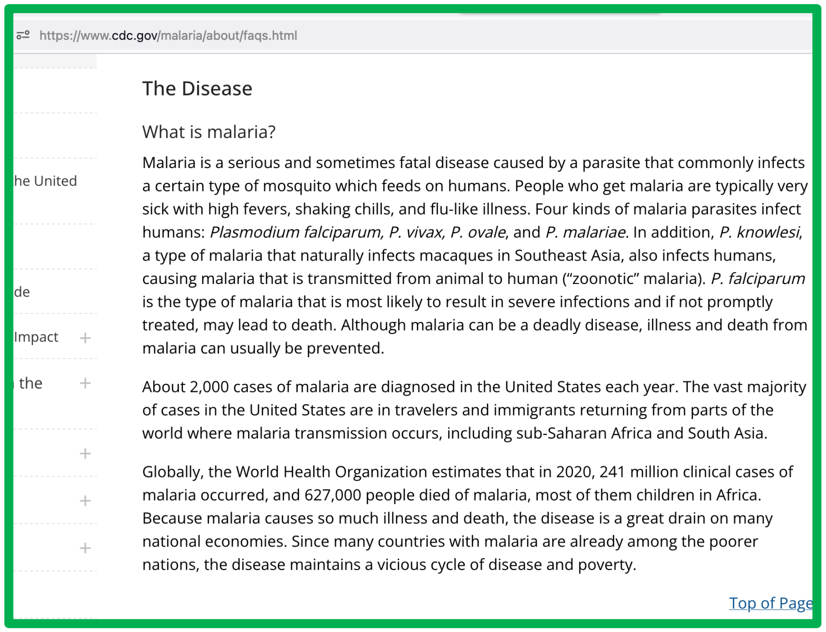
prog2processor –t “what is **malaria**?”” // Tell about disease

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.

…



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.

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., disease description comes after “The Disease”
  - 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

Concluding Section

Lecture 12: Concluding Comments

- We reviewed and tested Home Work #3 by peers
- Looked again at the concept of inheritance; covered inheritance type
- Looked again at the concept of polymorphism
- Looked again at Regex
- Discussed PA#2 and regex

About Next Lecture – Lecture 13

Lecture 13: Exceptions

- Helps handle runtime failures
- Benefits from using OO concepts

12	**Feb 17 (Th)**	**Review: inheritance, Polymorphism**	**HW 3 due**
13	Feb 22 (Tu)	Exceptions	Prog 2 - end
	Feb 24 (Th)	OO – Constructor, Destructor	Prog 3 - start
14	Mar 1 (Tu)	OO – operators, access control	HW 4 due
15	Mar 3 (Th)	C++ standard library	Prog 3 - end Semester - Midpoint