

CSCE 240: Advanced Programming Techniques

Lecture 12: Review Object Oriented Concepts – Inheritance, Polymorphism

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

17TH FEBRUARY 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 12

- Introduction Section
 - Recap of Lecture 9
 - TA and SI Updates
- Main Section
 - Code peer review and testing: Home work #3
 - Review: Inheritance
 - Review: Polymorphism
- Concluding Section
 - About next lecture – Lecture 13
 - Ask me anything

Introduction Section

Recap of Lecture 11

- Quiz 1
 - Feedback and discussion

Announcements

- Programming Assignment #1: marks posted

PA: Code **Reviewing** Rubric Used

- Look out for
 - Can one understand what the code is doing ?
 - Can one explain the code to someone else (non-coder) ?
 - Can one 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 (out of 100 -/+)

- -100: code not available
- -80: code with major issues
- -60: code with minor issues
- -20:
- 0: (full marks): no issues
- +20: special features

PA: Code **Testing** Rubric Used

- 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 ?
 - Is there a hardcoding of directory ? Paths should be relative to code base directory.
 - Any special feature?
- What not to judge
 - Length of documentation. It can just be short and accurate.
 - Person writing the code

Assign rating (out of 100 -/+)

- -100: code not available
- -80: code with major issues (e.g., abnormal termination, incomplete features)
- -60: code with minor issues
- -20:
- (full marks): no issues
- +20: special features

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 usecases

Updates from TA, SU

- TA update: Yuxiang Sun (Cherry)
- SI update: Blake Seekings

Main Section

Home Work 3

Due Thursday, Feb 17, 2022
- Originally due on Feb 10, 2022

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

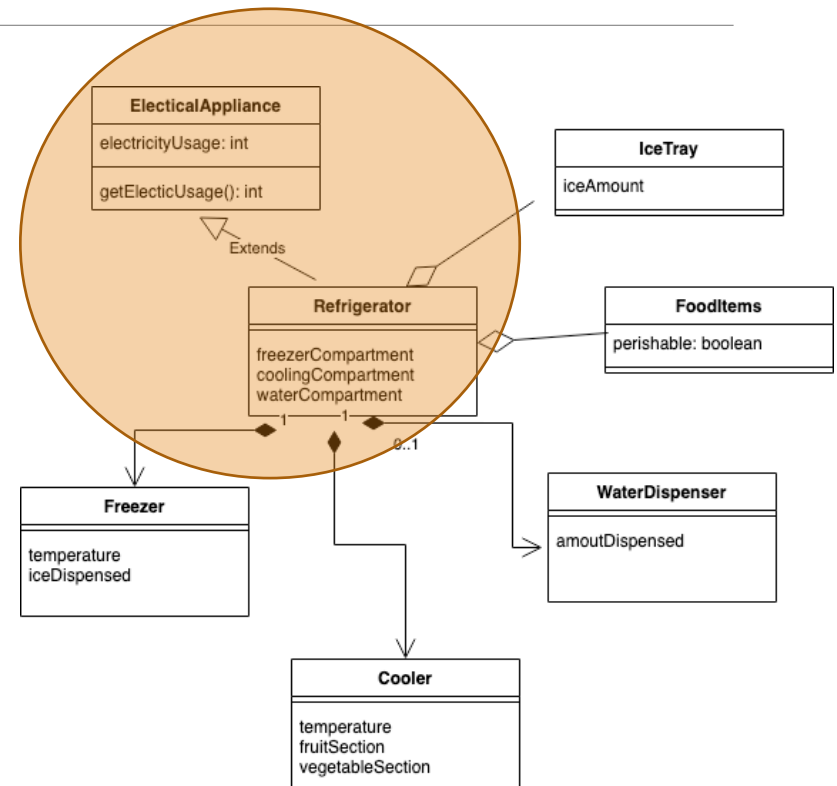
Review of Concept: Inheritance

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

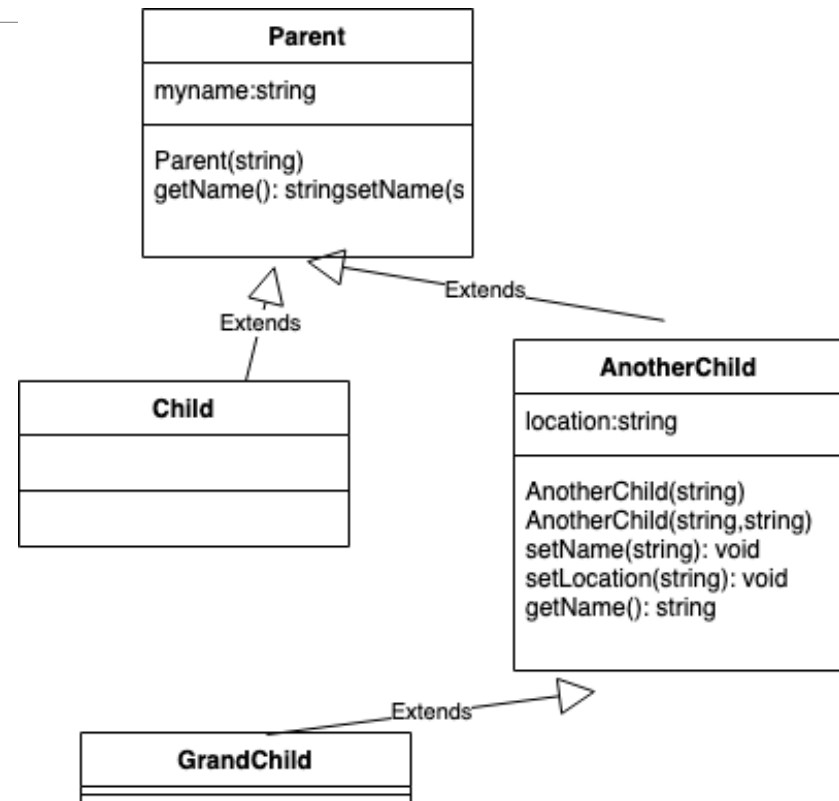
Why Use Inheritance ?

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



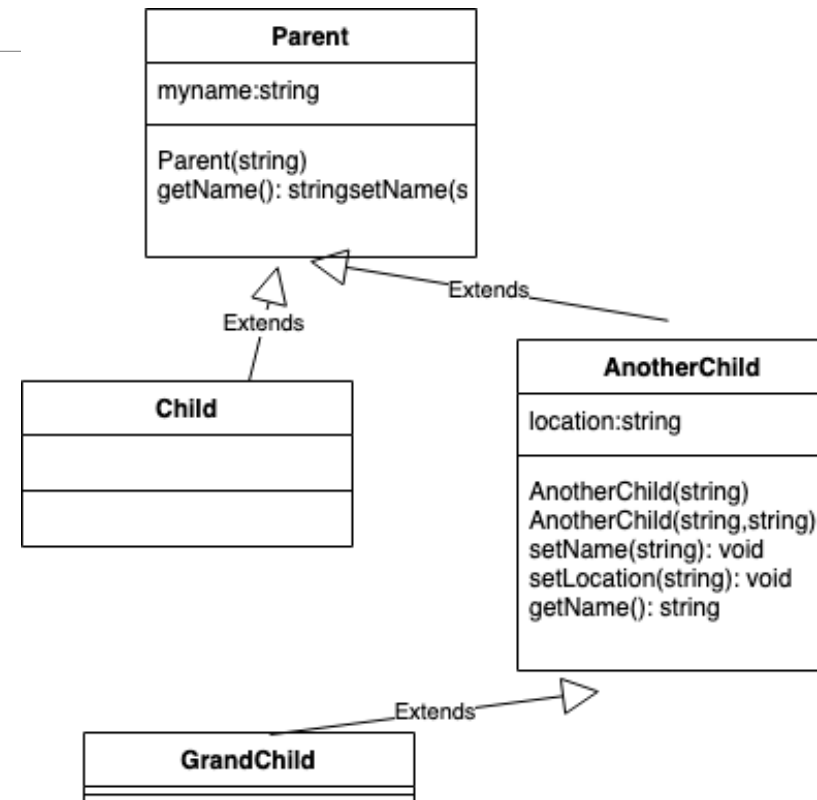
How to Use Inheritance ?

- 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

Review of Concept: Polymorphism

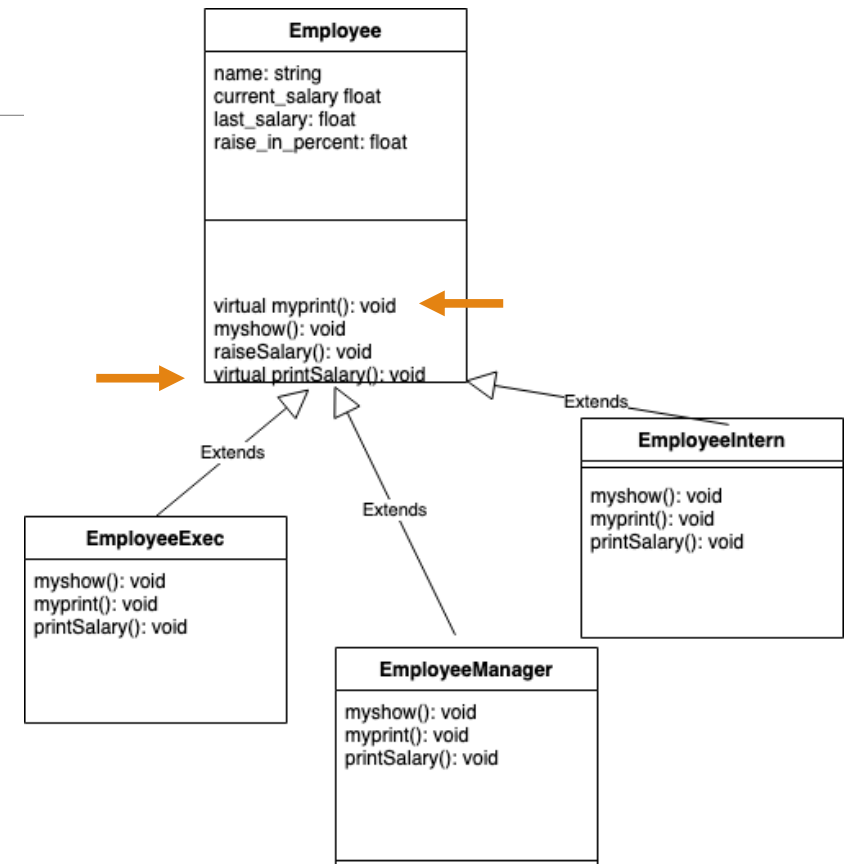
“Multiple shapes”

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**



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 ?

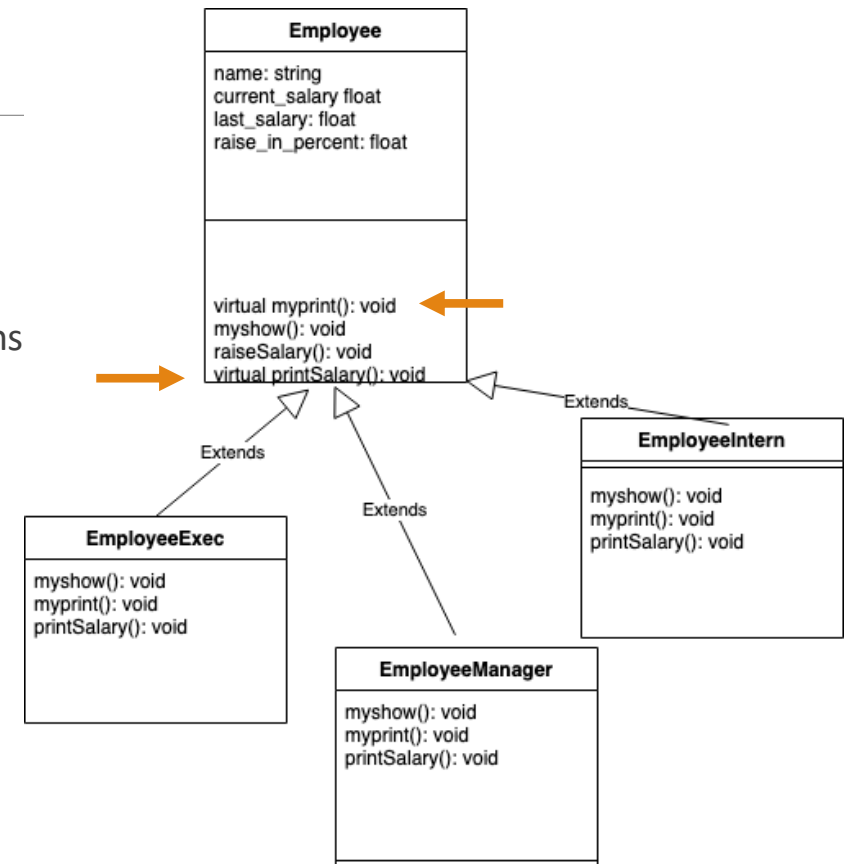
- Language independent syntax
- Illustration
 - 4 classes; 1 base, 3 derived
 - Basic: no data members; myshow() and myprint() functions
 - Advanced: 3 data members, printSalary() function

```
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

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

Discussion: Course Project

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
- **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 district name given as input)
 - Given an information type as input, the program will return its content
 - Examples: Contact Information, personal information, voting records
 - Input type can be given as command line argument.
Examples:
 - `prog2processor -t "Contact Information"`
 - `prog2processor -t "Contact Information:name" // Get person's name`
 - 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

- **Contact Information (Type-I1)**
 - Name
 - Region
 - Addresses: Columbia, Home
 - Phone: Business, Home
- **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)**

Example: Representative Information


Input:

```
prog2processor -t "Contact Information:name" // Get person's name
```

Output:

Terry Alexander

- 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: [Find Bills](#)

Voting Record

- Search Session: [Find Votes](#)

Service In Public Office

- Florence County Council, 1990-06, District Number 3
- House of Representatives, 2007 - Present

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

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

About Next Lecture – Lecture 13

Lecture 13: Quiz 1 (in class period)

- **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.

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