CS162, Lecture 11:
EXAM 1 NEXT WEDNESDAY

10/24/2018

15% of grade
Multiple Choice & True False
Next: Assignment 2
Hieroglyphics
Unit: $5
8. 5. 7. 10. 12. Ingredients:
1. Pepperoni
2. Cheese
3. Combo
4. Special price
5. Specify how many
6. Add another? y
7. Total money?
8. What size?
9. Order 1
10. Order 2

Place an order
Set up order
Select order has been selected

Prompt for customer info
Generate an order
Who is the customer?

array order sheet
Order Size
Quantity
Search pizza by ingredients

menu create a temp menu — dynamically grown
for each menu item
test included/omitted
add to the temp pizzas[]

return temp

Menu refined_menu = search_pizza_by_ingredients()
Menu a, b,
\[ a = b \]
Pizza (const Pizza & copy)  // copy constructor

Pizza ()

Pizza & operator=(const Pizza & copy)
for each pizza

for each ingredient in the current pizza

if pizza[ingredient] = ingredient
    add to refined menu
If time: Intro Inheritance
Inheritance

- The process by which a new class is created from another class
- Base (Parent) class == more general class which derived class are created from
- Derived (Child) class == new class
  - Has all of the member variables and functions as base class
- Examples:
Defining Inheritance

• class Derived: public Base {};
• In the Derived class:
  • List only member variables you want to add, not what is inherited
  • Only redefine inherited member functions if you want to redefine them
    • Redefining is when an inherited member function definition is changed in the derived class
• Derived classes can be used anywhere the base class would be used but not the other way around
Not Inherited

- Base class constructor - though it can be invoked from the derived class
  - Child::Child():Parent() {}
  - Base is called first to initialize all of the base member variables
  - If base constructor isn’t specified the base default constructor will be used
- Copy constructor
- Assignment Operator Overload
- Destructor
Inherited but Restricted

- Private member variables are inherited but cannot be accessed by name, need to use the accessor and mutator functions
- Private member functions are inherited but cannot be accessed by the derived class
- Protected access modifier allows for the derived class to be able to access things by name but every other class would view them as private