Students may use their textbook, any handwritten notes, any digital notes or programs written by the student, and the internet.

Monday, December 12th – 2016

6:00pm – 8:45pm – Be sure to attempt questions or programming tasks. Partial Credit will be awarded.

**SHORT ANSWER SECTION - Make sure to answer questions in your own words.** DO NOT COPY & PASTE FROM ONLINE RESOURCES.

96 Points

1. (5 Points) Explain the difference between Procedural Programming and Object Oriented Programming.

**In procedural programming, the program tells the computer what do do step by step. It is akin to following a recipe from top to bottom. Everything that needs to be done appears in order. In Object Oriented Programming, instead of step by step, programs are carried out by objects. Objects are made from classes. In the place of procedures, object methods carry out instructions. Every process to be completed does not appear in our main function. Some of them are in classes and objects elsewhere.**

2. (5 points) Describe Objects in table below:

|  |  |
| --- | --- |
| Class Objects | Other Objects Used in C++ |
| 1 Please - methods | 3 Comments - Array |
| 2 See - Class Variables | 4 Below - String |

**I do not understand the difference between a “class object” and an “other object.” I cannot think of a single object in C++ that does not have a class. There are some classes that we didn’t discuss, but everything I’ve read about *this* language, says that all objects are created from some class. For example, std::cout is an object from the ostream class and std::cin is an object created from the istream class (both objects being contained in the iostream header file). Even strings and arrays, as collections of primitive data types, have their own structs (which are very similar to classes) in their respective header files.**

How are objects created in C++, identify keyword syntax or object creation process.  
**Objects in C++ are created by first calling the class from which the object will be made, then naming the object. For example, if I had a class called Human and wanted to make an object called nigel, I would type:  
Human nigel;**

3. (5 Points) – Describe a “getter” method. Typically what type of method is used as a “getter” method? -2.5 Points Return Method

**A getter method is a function in a class. It usually retrieves a value stored in an object. For example, if an object carried out some calculations, and you wanted to see the values that were calculated, you could use a get method to access that attribute.**

4. (5 Points) – Describe a “setter” method. Typically what type of method is used as a “setter” method? -2.5 Points Return void  
**A setter method usually places values in an object. An object’s attributes may start blank when the object is created. To fill them, a set method is used.**

5. (10 Points) Explain iteration. How is iteration related to data structures and strings. Why is iteration important?  
**Data structures such as strings and arrays are related to iterations because they are a collection of primitive data types. They have a number of slots with indices that they fill with some primitive type such as a character or an integer. These objects are stored and retrieved through iteration. How? -5 pOints How does iteration allow the program to access elements in an Array.**

6. (10 Points) Describe the difference between class variables, local variables (Variables in a function that are not global), and global variables. Describe the process of assigning values to the different variable types, accessing the variables, and scope of each variable type. (In your own words.)

**Class variables and local variables have different scopes. A class variable exists for every derived class and object. It is related to that class and describes some aspect of it. Its scope is described at the time of its creation(public, private, or protected). Each aspect is accessed when objects are created from the classes, often through setter and getter methods. A local variable has the scope of the function in which it was created. It is meaningless outside of that function. It is accessed and modified inside of that particular function only.**

7. (10 Points) Describe Inheritance in your own words. Describe the key required syntax. (In your own words.)

**Inheritance is a quality of a class that allows it to pass on access of its public and protected variables and methods. If I were going to create a derivative class (class that will be inheriting attributes) of a base class (class passing on the information) I would type the following:**

**//Base class**

**class Shape {**

**public:**

**int length, width;**

**};**

**//Derived Class  
class Rhombus: public Shape: {**

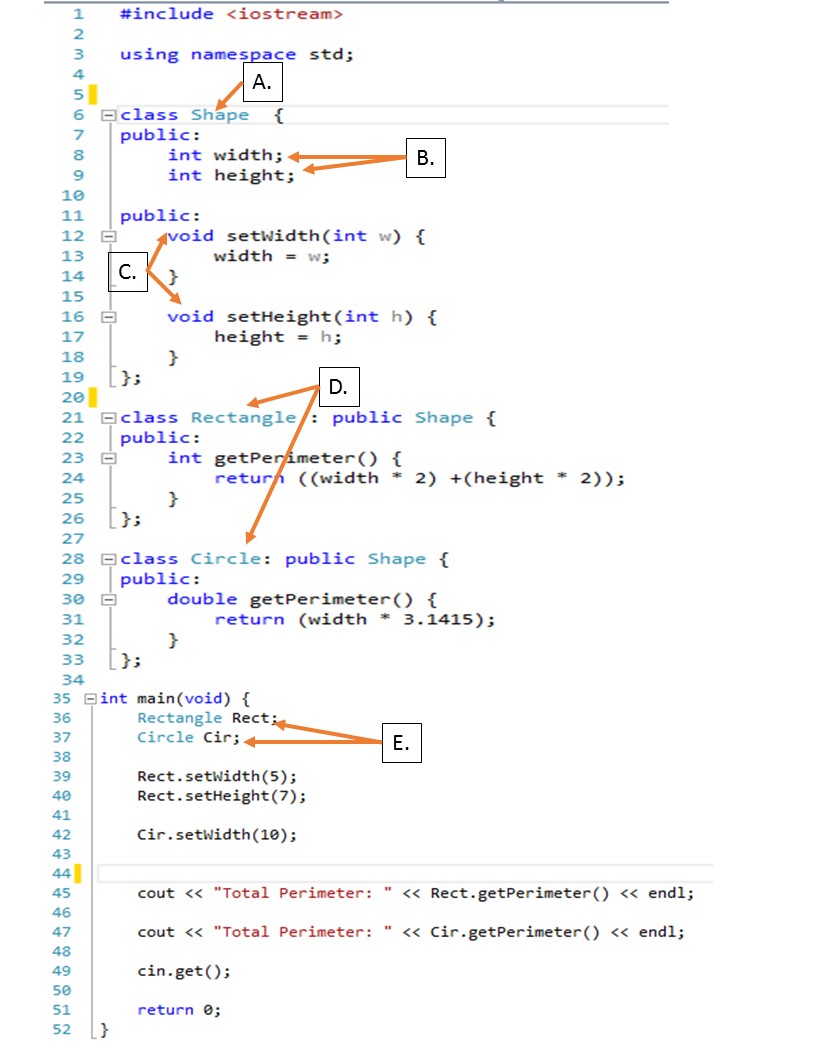
**};**

**The Derived class, *Rhombus*, will inherit the variables length and width. An object created from the *Rhombus* class will be able to access the length and width attributes with a dot operator.**

8. (10 Points) Compare and contrast Arrays and Pointers. (In your own words.)  
**Arrays and a pointer are similar. Many of the operations that can be done to pointers can also be done on arrays. Arrays can also be converted into pointers. The main difference between the two is that pointers may be assigned to different addresses and arrays are locked in the place in which they are created.**

**DIAGRAM PROGRAMMER CREATED CLASS:** (10 Points – 2 Points Each)

Name the class component and give a brief description in your own words the purpose it serves.



|  |  |
| --- | --- |
| A. | **A class called Shape is created. It is also the Base class from which later classes of Rectangle and Circle are derived**. |
| B. | **Class variables of integers called length and width are stored in the class for use by objects and derived classes** |
| C. | **Two public setter methods are added to the class Shape. It will be accessible by objects of that class and derivative classes** |
| D. | **Two derivative classes called Circle and Rectangle are created from the Base Class Shape. It inherits all the public attributes and would also inherit protected attributes if any were declared in the base class of Shape.** |
| E. | **Two objects called Rect and Circ are created from the classes Rectangle and Circle. The each inherit the aspects of the class from which they are created and the attributes of the Base Class Shape.** |

**SHORT PROGRAMMING TASKS:**

(30 points) TASK 2: Object Orientated Programming

Use Object Oriented Design (OOP),

Use Constructors, Parameters, and arguments as needed.

Create a program that can convert $dollars to Yen, Euro, or Peso.

The user will input a dollar value. If the $dollar value is greater than 0, create a menu that will allow user to select Yen, Euro, Peso, or all Currency conversion. If the $dollar value is less than 0 prompt the user and stop the program.

Write a program that has 4 methods that will return the concurrency conversions for the following.

Yen, Euro, Peso conversion program using a class structure.

Create 4 Methods:

1) Convert and output conversion in Yen

2) Convert and output conversion in Euro

3) Convert and output conversion in Peso

4) Convert and output all 3 currencies.

Attach Snipping photos of all 4 methods being executed.

Please enter $ Amount:

1000

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\* Main Menu: \***

**\* Enter # to run program or Quit \***

**\* 1) Yen Conversion \***

**\* 2) Euro Conversion \***

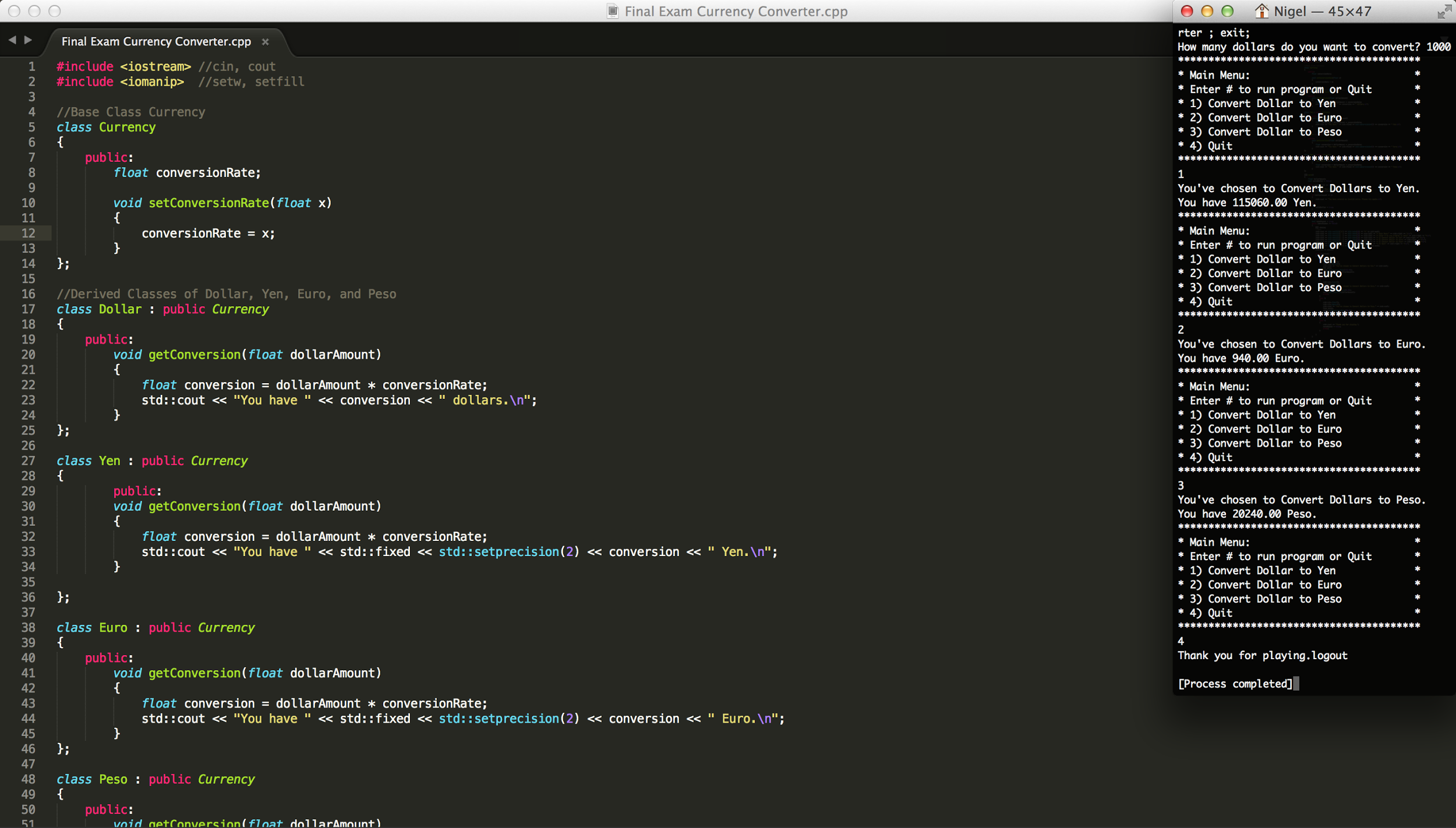
**\* 3) Peso Conversion \***

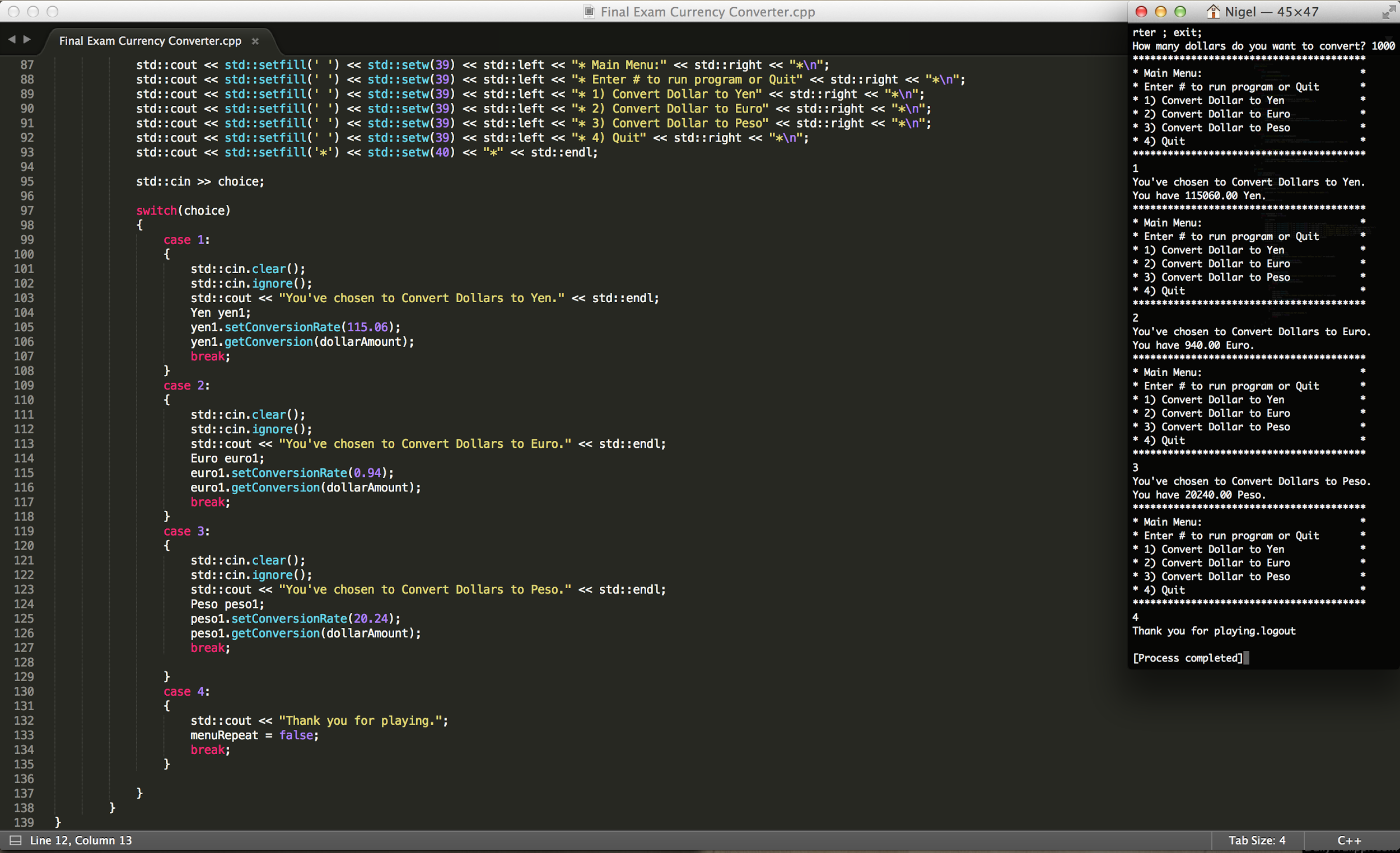
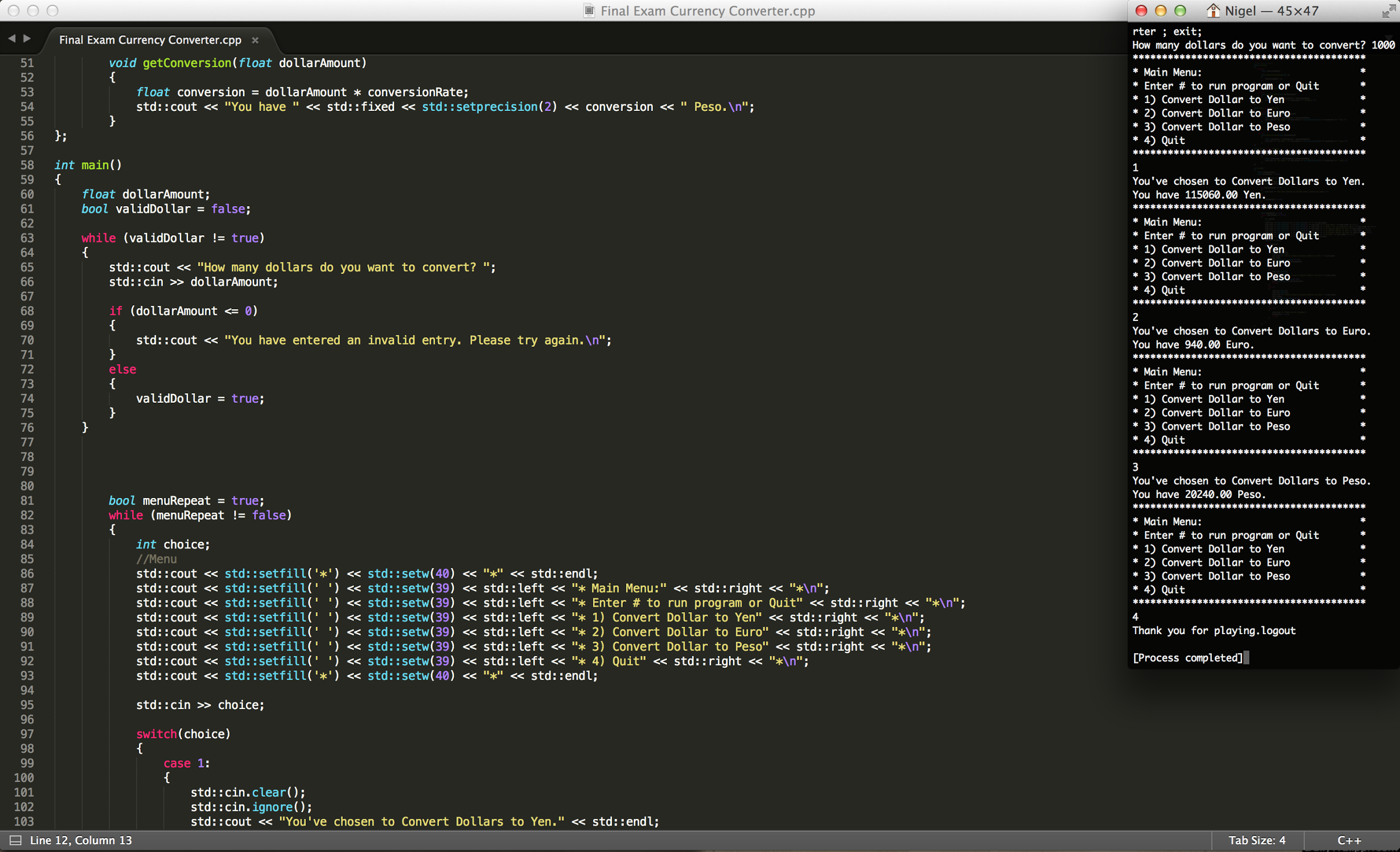
**\* 4) All Currencies \***

**\* 5) Quit \***

2

You have 943.40 Euros





(20 points) TASK 3:

Use **recursion** to **Draw ONE of the stare patterns below. Output any star pattern you like A,B, or C.**

**-10 points you do not use recursion. But a while Loop**

Attach Snipping photo of the source code and output below.

A.

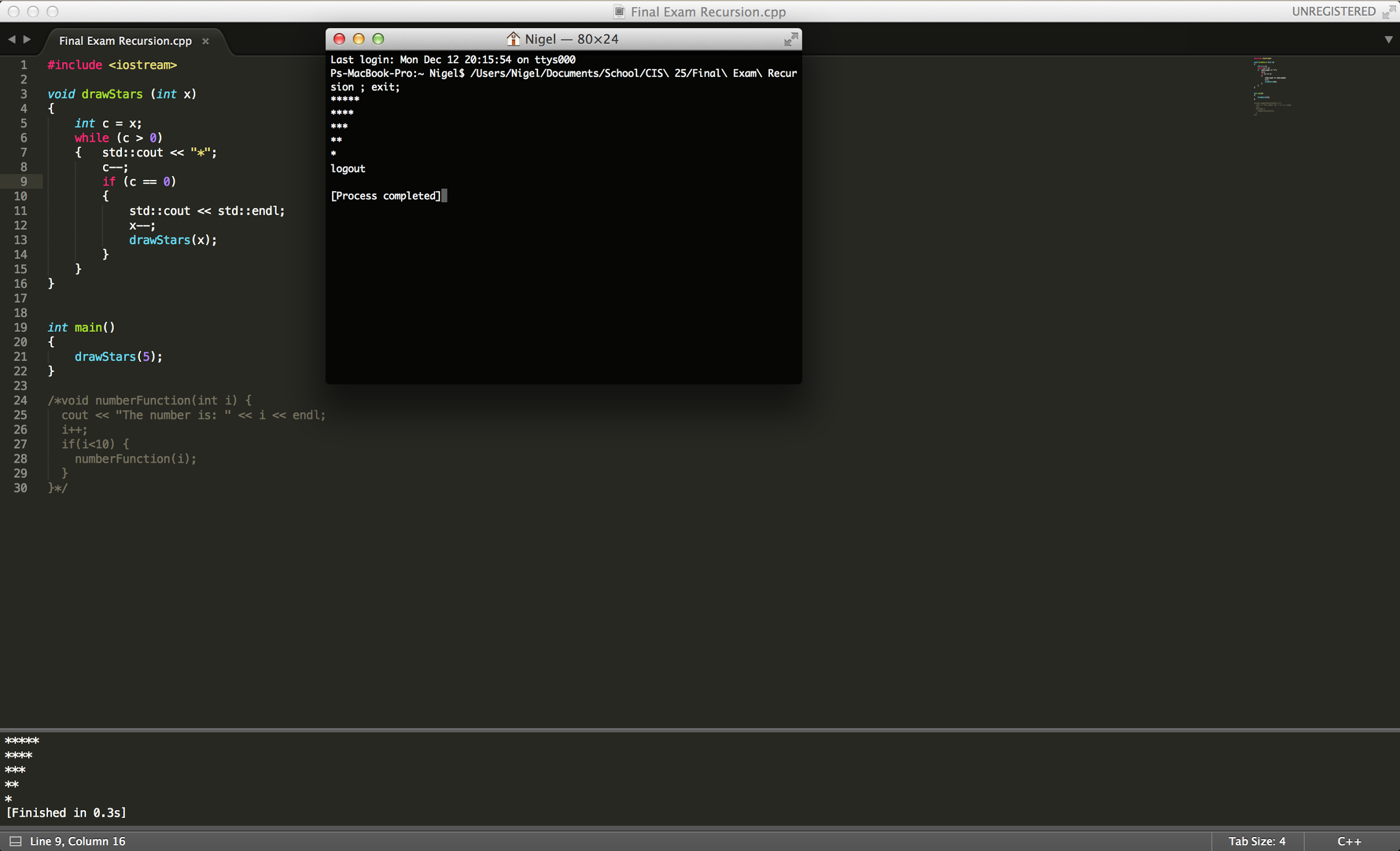
\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*



B.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

C.

\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*

\*\*