Lab 3 C++ I/O and SDLC

Introduction to Computer Science I

# Objectives:

After performing this lab, the students should be able to

* write C++ programs that involve input and output
* describe the phases of the Software Development Life Cycle

**Student Names:**

## Activity

Directions: For each programming exercise, provide your C++ source code and screenshot of your program output.

### Part I - ASCII Art

Write C++ programs to perform the following:

1. **Print a tree by adding a base under a 4-level triangle:**

**\***

**\*\*\***

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

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

**\*\*\***

1. **Print the following "cat":**

**/\ /\**

**o o**

**= =**

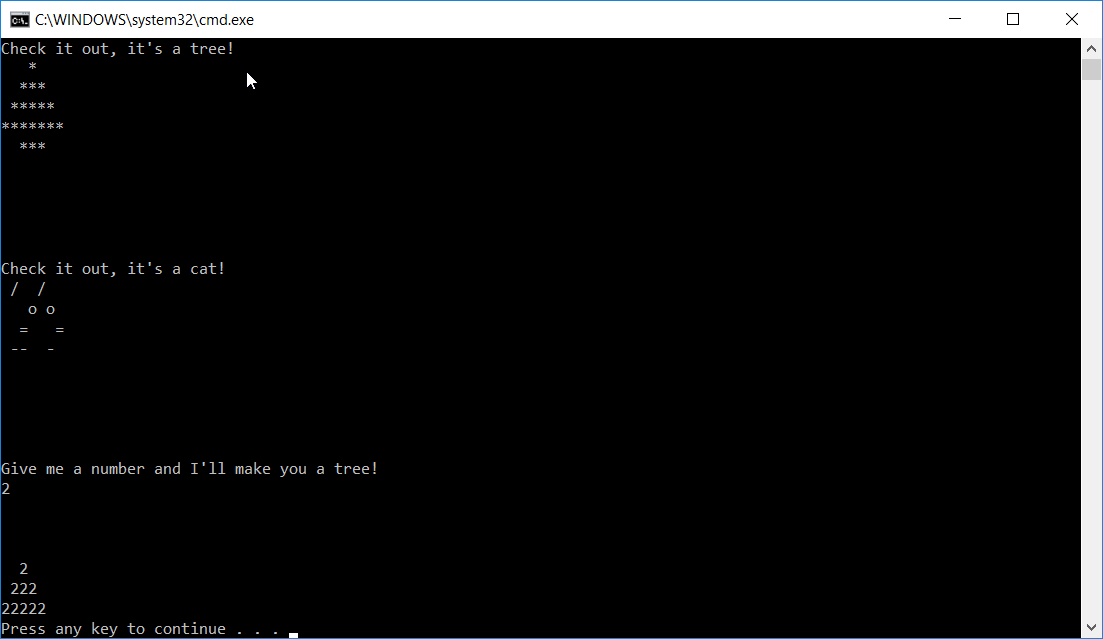
**---**

1. **Allow a user to enter a number, and then print a triangle using that number as in:**

**9**

**999**

**99999**



### http://upload.wikimedia.org/wikipedia/commons/thumb/1/19/SDLC_-_Software_Development_Life_Cycle.jpg/477px-SDLC_-_Software_Development_Life_Cycle.jpgPart II - SDLC

Read the first three sections of the chapter “Introduction to Software Engineering/Process/Life Cycle” on <http://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Process/Life_Cycle>.

In your own words, briefly explain each of the five phases of the SDLC:

1. **Analysis**

You take your project goals and break it down into defined functions. At this point you also review end-user needs. This is the “how would we do this?” stage.

1. **Design**

This is where you go deeper into the “How would we do this?” stage. This is where pseudo-code and other design tools come into play. i.e. wire frams, screen shots, mind mapping, etc.

1. **Implementation**

Time to write some code!

1. **Testing**

This is where you check your code. Checking for build errors, compile errors, etc. Usually ran through an automated build system. i.e. Jenkins, etc.

1. **Operations and Maintenance**

You thought you were done! You’re not. Bug fixes, enhancements, etc.