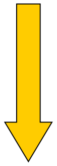
Program Development Life Cycle

- To be a successful Programmer requires
  - Analytical Problem Solving Skills
  - Troubleshooting Skills
  - Knowledge of Programming Language
  - Applying Program Development Life Cycle
- Program Development Life Cycle Phases
  1. Analysis Phase
  2. Design Phase
  3. Implementation Phase
  4. Use Phase



Top Down Development Process

1. Analysis Phase

- Investigate and analyze the nature of problem
  - Programmer meets with users to discuss and analyze the problem
  - What are the dependent inputs?
  - What are the desired outputs?
  - What processing will be required?
  - What are event triggers?
  - *Means-Ends Analysis*

- Write Program Specifications Report
  - Write the program requirements
  - Write the program specifications description
    - Describe the goals of the program
    - Describe appearance of input and output

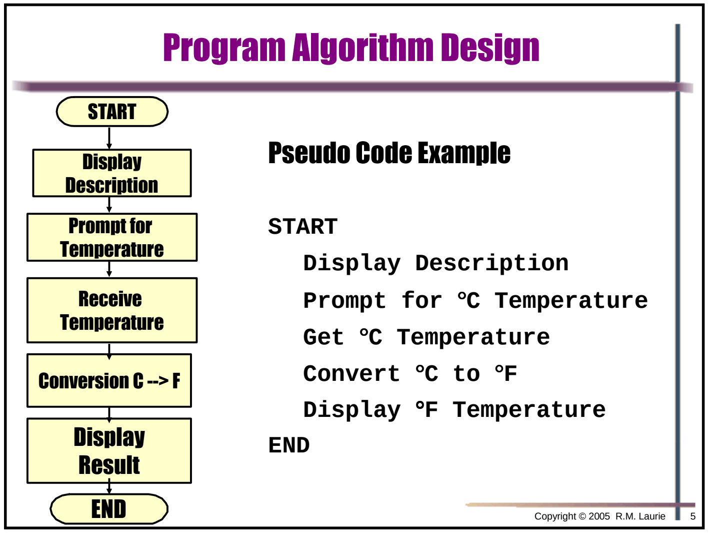
Program Specifications Report

- Program Requirements
  - Temperature conversion from degrees Celsius to degrees Fahrenheit
- Program Specifications Description
  1. The program will accept an entered Celsius temperature and convert it to Fahrenheit.
  2. Any real number may be entered.
  3. The appearance of input prompting and the results will be displayed as follows.

```
This program converts a temperature from
degrees Celsius to degrees Fahrenheit.

Enter the Celsius Temperature
>25   C

Results: 25C = 77F
```

2. Design Phase

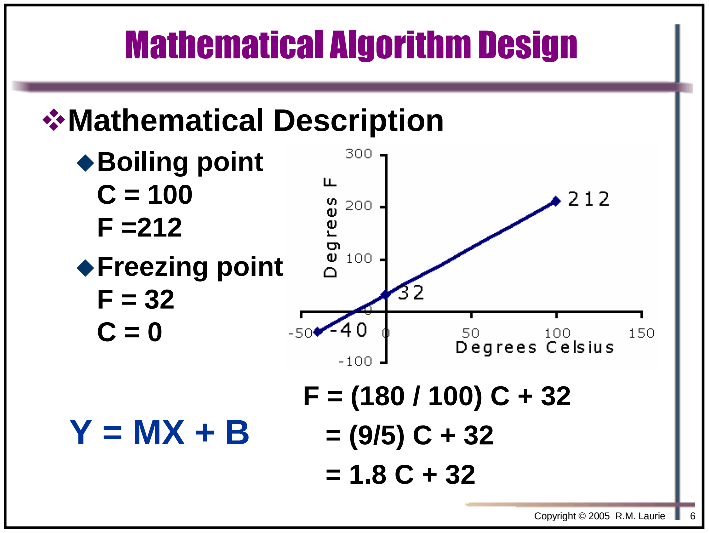
- Program Algorithm Design
  - Design program algorithms to achieve the desired results and model the required processing
  - *Flow Charts* - A pictorial representation of the ordered step by step process to solve the problem
  - *Psuedocode* - English like language that can state solution precisely but less less precision than C++
- Mathematical Algorithms Design
  - Perform mathematical analysis of problem
  - Determine mathematical equation algorithms
- Algorithms Verification
  - Desk check the algorithms by solving manually
  - Use known input test data and examine results to determine algorithm accuracy



Program Algorithm Design

Pseudo Code Example

```
START
   Display Description
   Prompt for °C Temperature
   Get °C Temperature
   Convert °C to °F
   Display °F Temperature
END
```

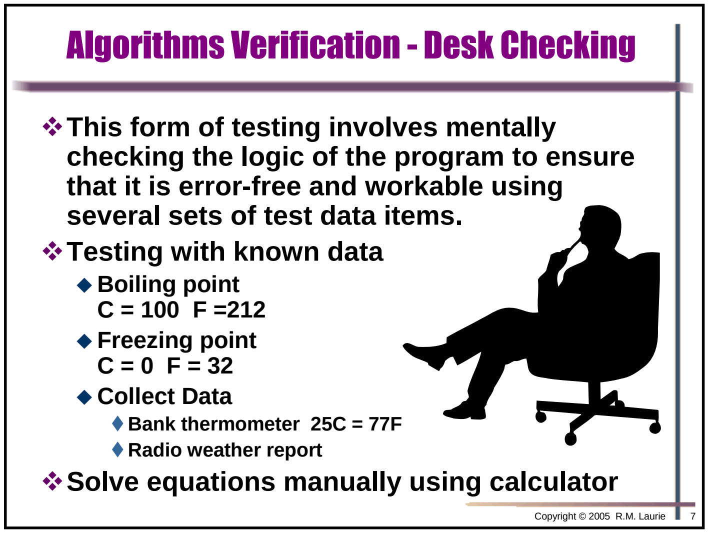


Mathematical Algorithm Design

- Mathematical Description
  - Boiling point
    C = 100
    F =212
  - Freezing point
    F = 32
    C = 0

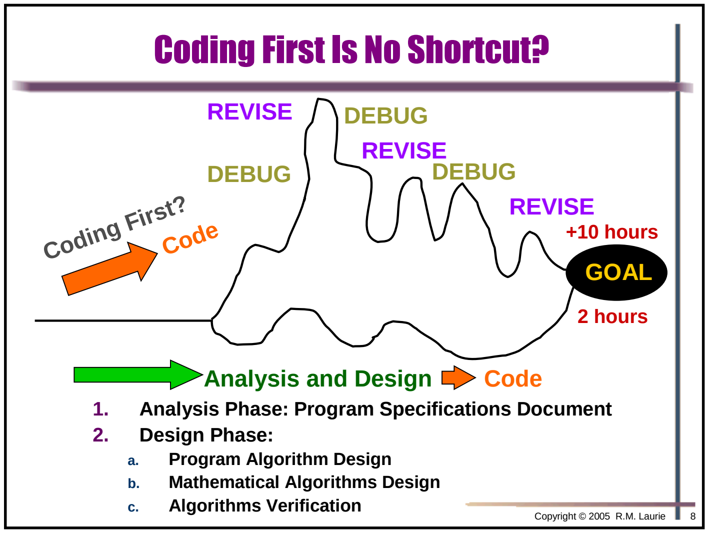
Y = MX + B

F = (180 / 100) C + 32
= (9/5) C + 32
= 1.8 C + 32



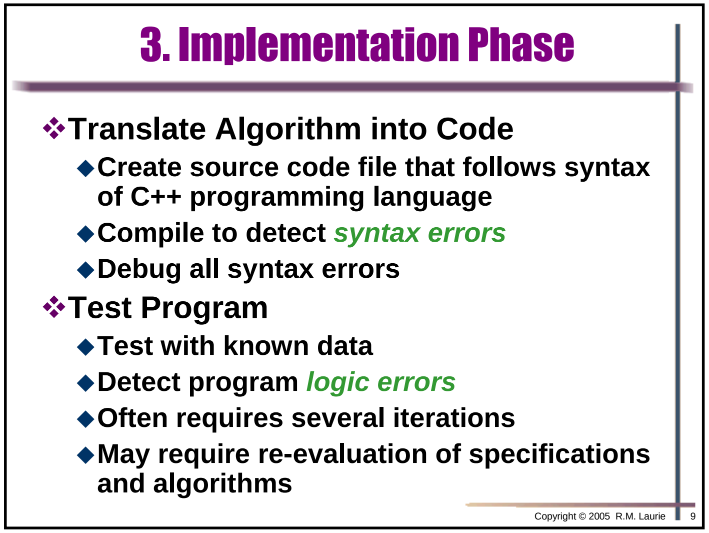
Algorithms Verification - Desk Checking

- This form of testing involves mentally checking the logic of the program to ensure that it is error-free and workable using several sets of test data items.
- Testing with known data
  - Boiling point
    C = 100 F =212
  - Freezing point
    C = 0 F = 32
  - Collect Data
    - Bank thermometer 25C = 77F
    - Radio weather report
- Solve equations manually using calculator



Coding First Is No Shortcut?

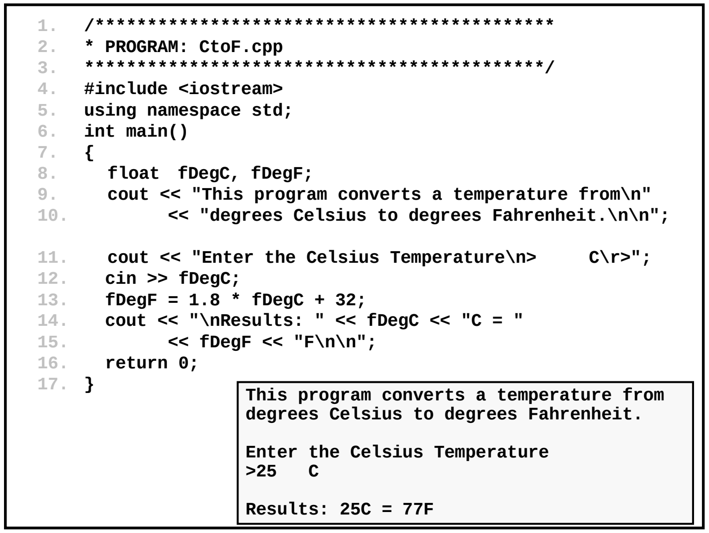
1. Analysis Phase: Program Specifications Document
2. Design Phase:
   a. Program Algorithm Design
   b. Mathematical Algorithms Design
   c. Algorithms Verification



3. Implementation Phase

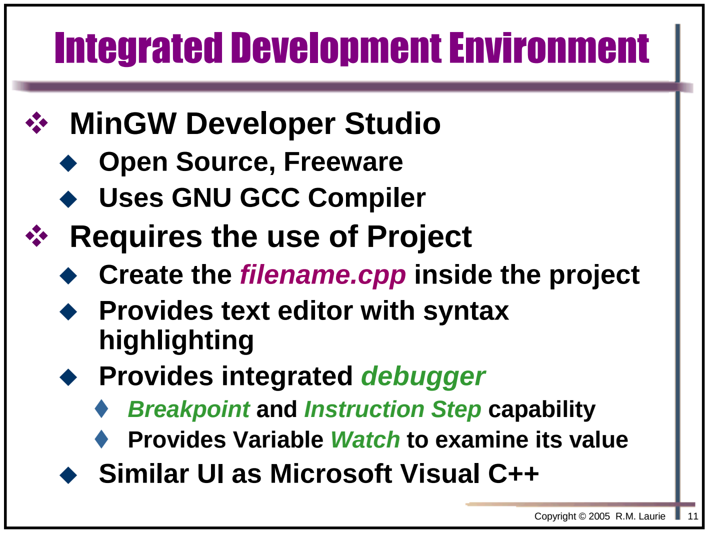
- ❖Translate Algorithm into Code
  - ◆Create source code file that follows syntax of C++ programming language
  - ◆Compile to detect *syntax errors*
  - ◆Debug all syntax errors
- ❖Test Program
  - ◆Test with known data
  - ◆Detect program *logic errors*
  - ◆Often requires several iterations
  - ◆May require re-evaluation of specifications and algorithms

Copyright © 2005 R.M. Laurie 9



```
/***********************************************
* PROGRAM: CtoF.cpp
***********************************************/
#include <iostream>
using namespace std;
int main()
{
  float  fDegC, fDegF;
  cout << "This program converts a temperature from\n"
       << "degrees Celsius to degrees Fahrenheit.\n\n";

  cout << "Enter the Celsius Temperature\n>     C\r>";
  cin >> fDegC;
  fDegF = 1.8 * fDegC + 32;
  cout << "\nResults: " << fDegC << "C = "
       << fDegF << "F\n\n";
  return 0;
}
```

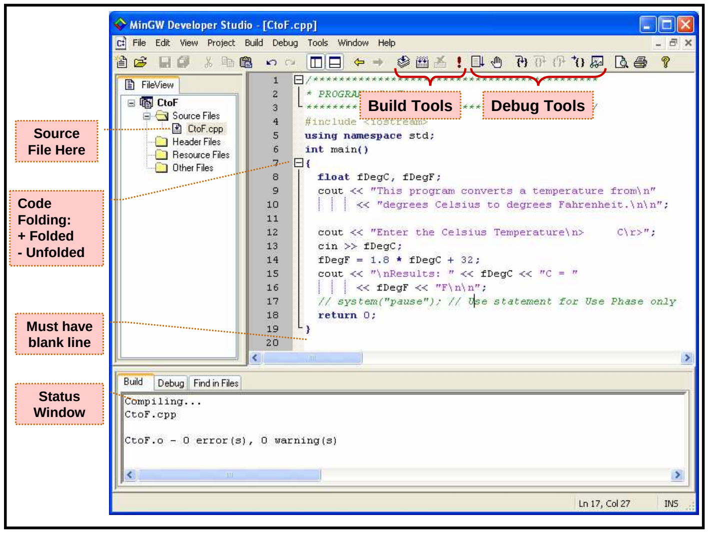
```
This program converts a temperature from
degrees Celsius to degrees Fahrenheit.

Enter the Celsius Temperature
>25   C

Results: 25C = 77F
```



Integrated Development Environment

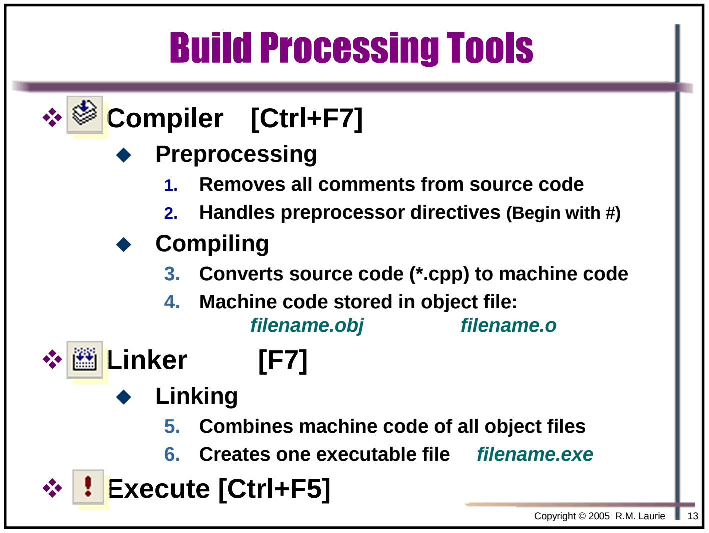
- ❖ MinGW Developer Studio
  - ◆ Open Source, Freeware
  - ◆ Uses GNU GCC Compiler
- ❖ Requires the use of Project
  - ◆ Create the *filename.cpp* inside the project
  - ◆ Provides text editor with syntax highlighting
  - ◆ Provides integrated *debugger*
    - ♦ *Breakpoint* and *Instruction Step* capability
    - ♦ Provides Variable *Watch* to examine its value
  - ◆ Similar UI as Microsoft Visual C++

Copyright © 2005 R.M. Laurie 11



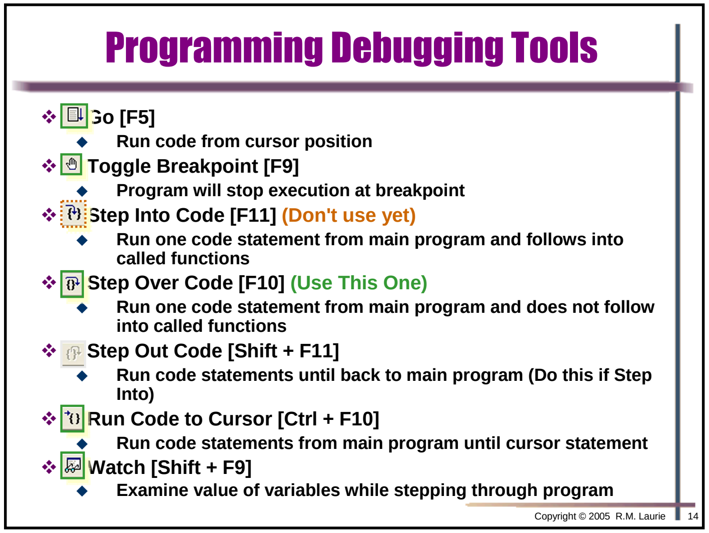
```
/****************************************
* PROGRA
*********
#include <iostream>
using namespace std;
int main()
{
   float fDegC, fDegF;
   cout << "This program converts a temperature from\n"
      << "degrees Celsius to degrees Fahrenheit.\n\n";

   cout << "Enter the Celsius Temperature\n>     C\r>";
   cin >> fDegC;
   fDegF = 1.8 * fDegC + 32;
   cout << "\nResults: " << fDegC << "C = "
      << fDegF << "F\n\n";
   // system("pause"); // Use statement for Use Phase only
   return 0;
}
```

```
Compiling...
CtoF.cpp

CtoF.o - 0 error(s), 0 warning(s)
```



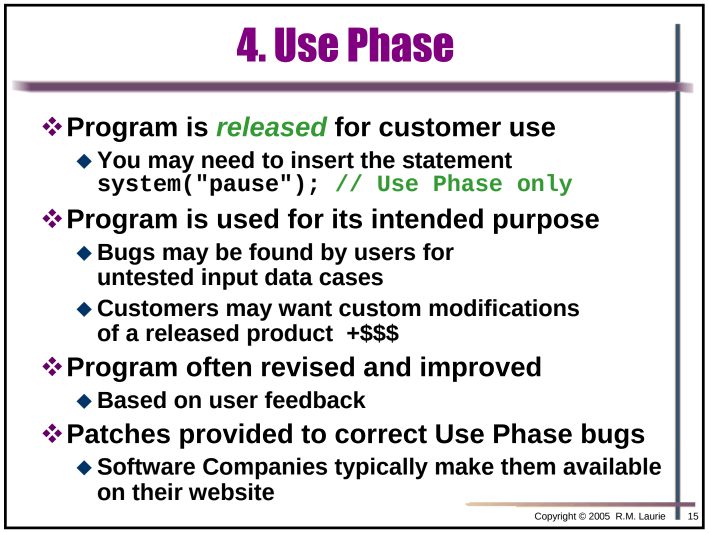
Build Processing Tools

- Compiler [Ctrl+F7]
  - Preprocessing
    1. Removes all comments from source code
    2. Handles preprocessor directives (Begin with #)
  - Compiling
    3. Converts source code (*.cpp) to machine code
    4. Machine code stored in object file:
       filename.obj *filename.o*
- Linker [F7]
  - Linking
    5. Combines machine code of all object files
    6. Creates one executable file *filename.exe*
- Execute [Ctrl+F5]



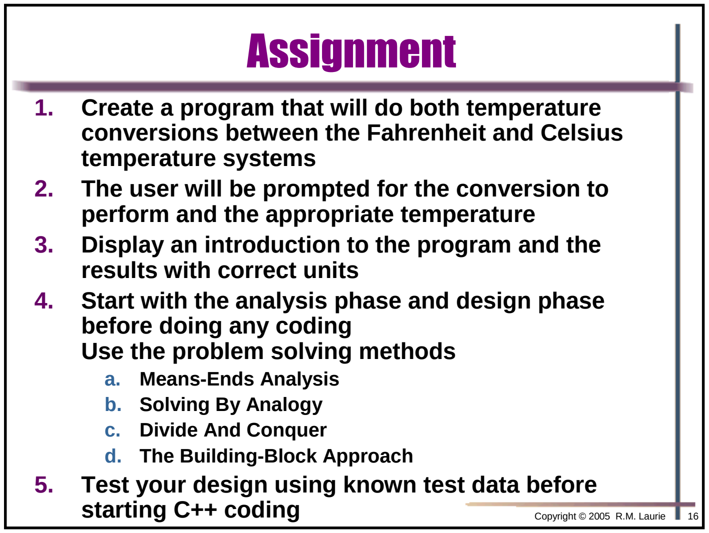
Programming Debugging Tools

- Go [F5]
  - Run code from cursor position
- Toggle Breakpoint [F9]
  - Program will stop execution at breakpoint
- Step Into Code [F11] (Don't use yet)
  - Run one code statement from main program and follows into called functions
- Step Over Code [F10] (Use This One)
  - Run one code statement from main program and does not follow into called functions
- Step Out Code [Shift + F11]
  - Run code statements until back to main program (Do this if Step Into)
- Run Code to Cursor [Ctrl + F10]
  - Run code statements from main program until cursor statement
- Watch [Shift + F9]
  - Examine value of variables while stepping through program



4. Use Phase

- Program is *released* for customer use
  - You may need to insert the statement
    `system("pause"); // Use Phase only`
- Program is used for its intended purpose
  - Bugs may be found by users for untested input data cases
  - Customers may want custom modifications of a released product +$$$
- Program often revised and improved
  - Based on user feedback
- Patches provided to correct Use Phase bugs
  - Software Companies typically make them available on their website



Assignment

1. Create a program that will do both temperature conversions between the Fahrenheit and Celsius temperature systems
2. The user will be prompted for the conversion to perform and the appropriate temperature
3. Display an introduction to the program and the results with correct units
4. Start with the analysis phase and design phase before doing any coding
   Use the problem solving methods
   a. Means-Ends Analysis
   b. Solving By Analogy
   c. Divide And Conquer
   d. The Building-Block Approach
5. Test your design using known test data before starting C++ coding