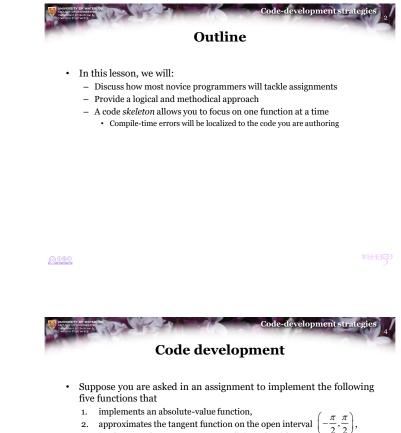




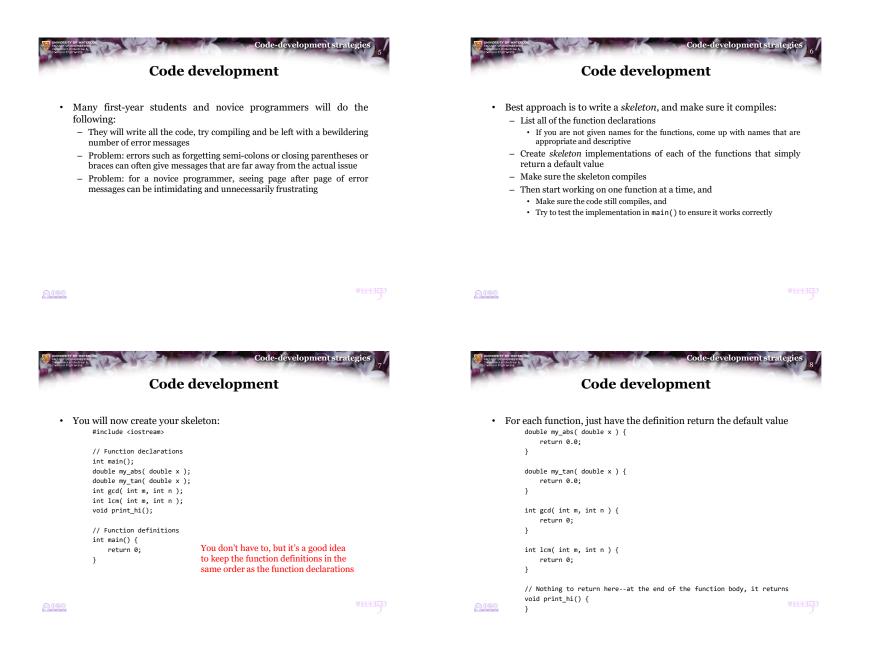
- · In the course assignments, you will be asked to implement a number of functions, and later, structures and classes
- · We have now looked at functions, conditional statements and repetition statements
 - We will now look at strategies for developing a code base

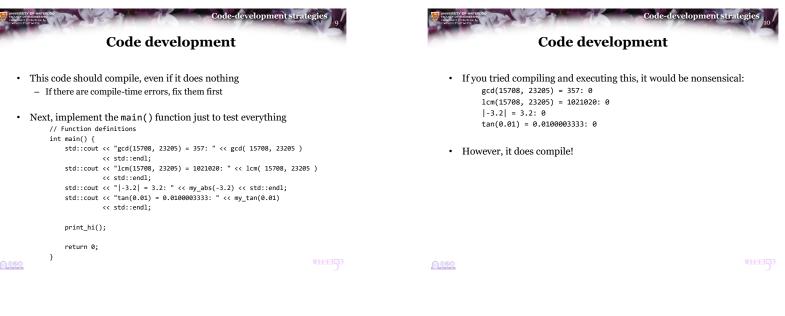


- approximates the tangent function on the open interval 2.
- implements the greatest-common divisor algorithm, 3.
- implements the least-common multiple algorithm, and 4.
- asks the user for a name, e.g., "Bob" and then prints "Hi Bob!". 5.

together with a main() function that uses these functions

1







- · Next, implement one function at a time, and make sure it works
 - Don't start lcm until you get gcd working
 - If you think that the absolute-value function is easier, start with it, but make sure it works before you go onto the next
- · Many students have an attitude combining thoughts like:

"This is too easy..."

"I won't make any mistakes!"

"I want to get this #\$@&%*! done and over with!!!" "Compiling wastes my time..."

- · And one in one hundred students will be right on all counts
 - The rest, however, may ignore these suggestions and in the end, waste more time than they would have had to... ${\color{red} { \pmb \otimes }}$

FCE15Ø



- · Following this lesson, you now:
 - Understand that writing everything at once is likely going to result in longer development time
 - Know that starting by writing a skeleton is easy enough
 - Understand that by writing one function at a time, you can
 - · First, focus on any compile-time errors your implementation has
 - Next, ensure your implementation works correctly
- The course web site presents an example that prints the volume and surface areas of the five Platonic solids:

https://ece.uwaterloo.ca/~ece150/Lecture_materials/1.10/



ECEIEØ



[1] Wikipedia https://en.wikipedia.org/wiki/Skeleton_(computer_programming)



Proof read by Dr. Thomas McConkey and Charlie Liu.





These slides were prepared using the Georgia typeface. Mathematical equations use Times New Roman, and source code is presented using Consolas.

The photographs of lilacs in bloom appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens on May 27, 2018 by Douglas Wilhelm Harder. Please see

https://www.rbg.ca/









These slides are provided for the ECE 150 *Fundamentals of Programming* course taught at the University of Waterloo. The material in it reflects the authors' best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.

0000

