Setting the precision of output

To set the precision of floating point output to "n" decimal places, use the following three commands:

```cpp
cout.setf(ios::fixed); // Set output to fixed precision
cout.setf(ios::showpoint); // Show decimal point
cout.setf(ios::fixed); // Optional: Ensure fixed precision
```

See sample program `payrate.cpp` on the website.
The typedef statement allows you to define a new variable type based on an existing type.

For example, the line `typedef float RealType;` allows you to define a `RealType` variable with the command `RealType x;`.

Using a typedef statement of this form allows you to easily change the type of all the `RealType` variables in your program.

The typedef statement allows you to define a new variable type based on an existing type.
The typedef statement

- If you decide at some later date that your program needs more precision, you can simply change the typedef line to
  ```
  typedef long double RealType;  
  typedef double RealType;  
  ```

- See the sample programs radius-float.cpp and radius-double.cpp on the website.

- Place the typedef line in the area where you define global constants, outside of the main function.

- If you decide at some later date that your program needs more precision, you can simply change the typedef line.
The assert function

- To aid your debugging, you may want to test each function’s preconditions with an assert statement.
- To use the assert function you need to include the cassert library.
- The syntax is `assert(boolean expression);`
- See the sample program `factorial.cpp` on the website.
Strategic cout statements:

- Place `cout` statements in your program inside complicated loops or inside conditional expressions while debugging.
- Remove these statements when your program runs correctly.

Strategic cout statements
Classwork/Homework

- Experiment with the sample programs associated with today's lecture:
  - `payrate.cpp`
  - `radius-float.cpp`
  - `radius-double.cpp`
  - `factorial.cpp`

- Revisit the sorting function from Exercise #2, add cout statements to the sort function to help you understand what is happening as the sort function runs. You might want to output the entire array at each step of the sorting loop.

- With today's lecture: payrate.cpp, radius-float.cpp, radius-double.cpp.