Solving the Problem

- **Problem solving phase**
  - Define the problem
  - Come up with an algorithm to solve the problem
  - Test the algorithm
- **Implementation phase**
  - Translate algorithm to programming language
  - Test the program

Top-down design

- Break problem into several logical subtasks
- Break sub-tasks into smaller tasks, and so forth.

Variables

- A variable sets aside memory for data and gives that place a name (identifier).
- Always has a value (even if garbage)

Scope

- Global constants (or variables) are declared outside of main.
- Local variables (or constants) are declared inside body of function or main.

Flow of Control

The order in which statements are executed.

- Sequential
- Branching – chose between alternatives:
  if statement and if/else statement
  switch statements
- Looping – repeat a statement or group of statements:
  while statement and do while statement

Loops

For :

```
for (init statement; condition; update)
{
}
```

Run looptest.cpp

Do – While:

```
do
{
}
while (condition is true)
```

While:

```
while (condition is true)
{
}
```

Functions

- **Pre-defined functions** (need to include library)
- **Programmer-Defined Functions**

Definition is done in two parts:

- Function Prototype (what does it look like)
- Function Definition (how does it do it)

(can be void functions that do not return values)

Passing parameters to functions

- Call-by-Value Parameters
- Call-by-Reference Parameters

Programming Style

- Grouping things that belong together:
  - Indenting (notice the C++ editor does some of this for you)
  - Leaving a blank line
- Comments
- Give explanatory names to variables

Debugging

- Start with first error (all others could be cascade errors)
- Error might be on previous line
- Watch spelling and semicolons
- After program is running, must do check for correctness
- Just because the program ran without errors does not mean the output is correct!