Scope

- Scope is a region of the program. There are three places, where variables can be declared.
- **Local**. Inside a function or a block which are called local variables, e.g.

```
int main()
{
      int i = 5;      // i created and initialized here
      float j = 4.0;  // j created and initialized here

      return 0;
} // i and j go out of scope and are destroyed here
```

- Variables defined inside nested blocks are destroyed as soon as the inner block ends, e.g.

```
int main() // outer block
{
      int i = 5; // i created and initialized here

      { // begin nested block
            float j = 4.0; // j created and initialized here
      } // j goes out of scope and is destroyed here

      // j can not be used here because it was already destroyed!
      return 0;
} // i goes out of scope and is destroyed here
```

- **Formal**. In the definition of function parameters which is called formal parameters. Here, the `seconds` variable has scope within the `pause()` function.

```
void pause(int seconds);
```

- **Global**. Outside of all functions which are called global variables, e.g.

```
// Variables declared outside of a block are global variables
int gX;     // global variable gX
int gY = 2; // global variable gY

void doSomething()
{
      // global variables can be seen and used everywhere
      gX = 3;
      cout << gY << "\n";
}

int main()
{
      // global variables can be seen and used everywhere
      gX = 5;
      cout << gY << "\n";
      return 0;
}
```

- Typically global variables are named starting with a lower case g, except if a `const`.