# Lecture 2. **Debugging & Troubleshooting**

**Troubleshooting** - Process of identifying and resolving a problem, error or fault within a software or a computer system. ("What is Troubleshooting”)

Types of errors:

**Compilation errors** – Errors reported when the code is compiled.

#### Compiler warnings

* Will not prevent the code from being compiled
* Should not be ignored since they often indicate a possible runtime or logic error (not otherwise easy to detect)

#### Compiler errors

* Must be fixed for the code to be compiled
* Result of syntax errors
  + **Syntax errors** - Code does not conform to the syntax of the programming language and the compiler cannot understand it
  + Examples: extra bracket, missing semicolon
* Compiler errors normally include a line number at which the error is found

**Linker errors** - Occur at the linking stage

* Indicate that the code compiles but functions/library/global variables that program is using are declared but not defined

**Logic errors** - Cause a wrong program's output/results

* Indicate that the code contains some incorrect logic, i.e. "bugs". These errors are completely preventable.

**Runtime errors** - Occur during the execution of the program

* Caused by wrong user input, hardware failures, such as a lack of resource (e.g., memory space, disk space, etc.), network connection failure, some mathematical operations (e.g., division by 0, square root of a negative number, etc.).

**Debugging**

* Process of finding and removing computer program errors, called “bugs”.
* Errors must be corrected to allow proper program execution.

Debugging Techniques ("Debugging with Visual Studio 2005/2008, Part 1: Debugging Concepts"):

* Local debugging
* Remote debugging
* Breakpoints
* Trace output
* Dump files
* Log files

## References

* "What is Troubleshooting? - Definition from Techopedia." *Techopedia.com*. N.p., n.d. Web. 13 Apr. 2017. <<https://www.techopedia.com/definition/5574/troubleshooting>>.
* "What is Debugging? - Definition from Techopedia." *Techopedia.com*. N.p., n.d. Web. 13 Apr. 2017.

<<https://www.techopedia.com/definition/16373/debugging>>.

* "Dealing with Compiler Errors - Surviving the Compilation Process." *Dealing with Compiler and Linker Errors - Cprogramming.com*. N.p., n.d. Web. 13 Apr. 2017.

<<http://www.cprogramming.com/tutorial/compiler_linker_errors.html>>.

* "Debugging with Visual Studio 2005/2008, Part 1: Debugging Concepts." *Debugging C using Visual Studio 2005/2008: Overview of Debugging Concepts - Cprogramming.com*. N.p., n.d. Web. 13 Apr. 2017. <<http://www.cprogramming.com/tutorial/debugging_concepts.html>>.

# Appendix 1: Using MS Visual Studio - Running & Troubleshooting a Program

* To compile a program, click on **Build** 🡪 **Build Solution** (new program) or **Build** 🡪 **Rebuild Solution** (program previously compiled)
* To run a program continuously (and debug), click on **Debug** 🡪 **Start Debugging** OR click on  (below **Tools**)

### To run a program one line at the time, click on **Debug** 🡪 **Step Into** (or **F11**) and then click on **Debug** 🡪 **Step Over** (or **F10**) every time you want to execute one line of the code. Yellow arrow points to the next statement to be executed.

* To monitor selected variables while executing a program one line at the time, click on

**Debug** 🡪 **Windows** 🡪 **Watch** 🡪 **Watch 1** (or 2/3/4). **Watch 1** window appears.

### Move the cursor to **Watch 1** window and enter names of variables you want to monitor. NOTE: You have to be in debug mode in order to open the **Watch** window. The proper sequence is:

* 1. Step into program (**Step Into** – **F11**)

### Setup **Watch** window and enter variables to monitor

* 1. Execute one line at the time (**Step Over** – **F10**) while monitoring variables.
* To stop executing/debugging a program, click on **Debug** 🡪 **Stop Debugging**.