# Exercises: Debugging

This document defines the **in-class exercises** assignments for the ["High-Quality Code" course @ Software University](https://softuni.bg/courses/high-quality-code).

## Fix Solutions using Debugging Techniques

Your task is to read the assignments and solutions for the problems given below. **Debug** and **fix any bugs** you encounter so that the solution passes the Judge tests. You may **NOT** rewrite the solution - practice your debugging skills!

* Judge System Contest: <https://judge.softuni.bg/Contests/Practice/Index/85#0>

Some important tools you need to look at are:

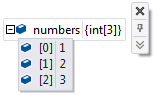
**Debugging workflow:**

* Start Debugging (**Debug > Start Debugging**, **F5**)
* Stop Debugging (**Debug > Stop Debugging**, **Shift + F5**)
* Step Over – executes the current line of code and goes on to the next (**Debug > Step Over**, **F10**)
* Step Into – if the current line contains a method, the debugger continues inside it (**Debug > Step Into**, **F11**)
* Step Out – if the debugger has stepped into a method, it leaves it; it does the reverse of Step Into (**Debug > Step Out**, **Shift + F11**)
* Toggle Breakpoint – adds / removes a breakpoint from the current line in the code (**Debug > Toggle Breakpoint**, **F9**)

**Debugging Windows:**

* Autos – shows all variables in the previous and the current line of code (**Debug > Windows > Autos**)
* Locals – shows all local variables in the current method (**Debug > Windows > Locals**)
* Watch (1…4) – shows the values of various expressions (**Debug > Windows > Watch > Watch 1…4**)
* Call Stack – shows the execution stack of the program (**Debug > Windows > Call Stack**)
* Breakpoints – shows all lines of code which have breakpoints (**Debug > Windows > Breakpoints**)
* (optional) IntelliTrace – allows to see what happened during a previous debugging session (**Tools > Options > IntelliTrace > General > Enable IntelliTrace**)
* (optional) Disassembly – shows the assembly code generated by the compiler (**Debug > Windows > Disassembly**)
* (optional) Memory / Registers – show the memory and CPU registers states ((**Debug > Windows > Memory / Debug > Windows > Registers**)

**Inline variable inspection:** hover over any variable in the current context and view its value. You can also "pin" a variable (using the pin icon) so you can see it along with the code.



The keyboard shortcuts are valid for the default key mapping scheme in Visual Studio and may differ from one environment to another.