EXPERIMENT 3*
CONTINUATION OF LAB 2 C PROGRAMMİNG

Aims

1. Learning primary functions of Microsoft Visual Studio
2. Introduction to C Programming
3. Running C programs using Microsoft Visual Studio
4. Debugging C programs using Microsoft Visual Studio

(Attention please: step 1-5 are included in Lab2, Lab3 starts from step 6)

6. Unassigned variable

Warning: Variable ... is used without initializing

This occurs if the given variable is not initialized before it is used. In C programming language the initial value of variables are undefined. Therefore, you have to assign a value to it before you use. This problem causes a warning to be issued and you can tell it to continue. However, your program might have problems in run time.

7. Address Error

Error: System error or access denied

These errors are mostly related with “scanf” function. If you want to read a variable in this function, you may need to use address operator (&).

Debugging

There are programs to find logic errors in your programs or help you to understand how your program works. These programs are called debuggers. A debugger can be a part of an integrated development environment. Microsoft Visual Studio also has its own debugger integrated into itself. This program starts working as soon as you start your program. When your program is paused or interrupted debugger will provide you information about your program flow, variables. Also it allows you to run your program step by step.

You can use two methods to pause your program. First method is to choose “Step over” (F10) from the debug menu to run your program. If you use this method your program will pause at the first line. The second method is to break your program execution when it reaches to a certain point. This eases the debugging of large programs. To add a “Break point”, choose “Toggle break point” from “Debug” menu. After running your program, it will stop at the break point you have specified. You may place multiple break points.

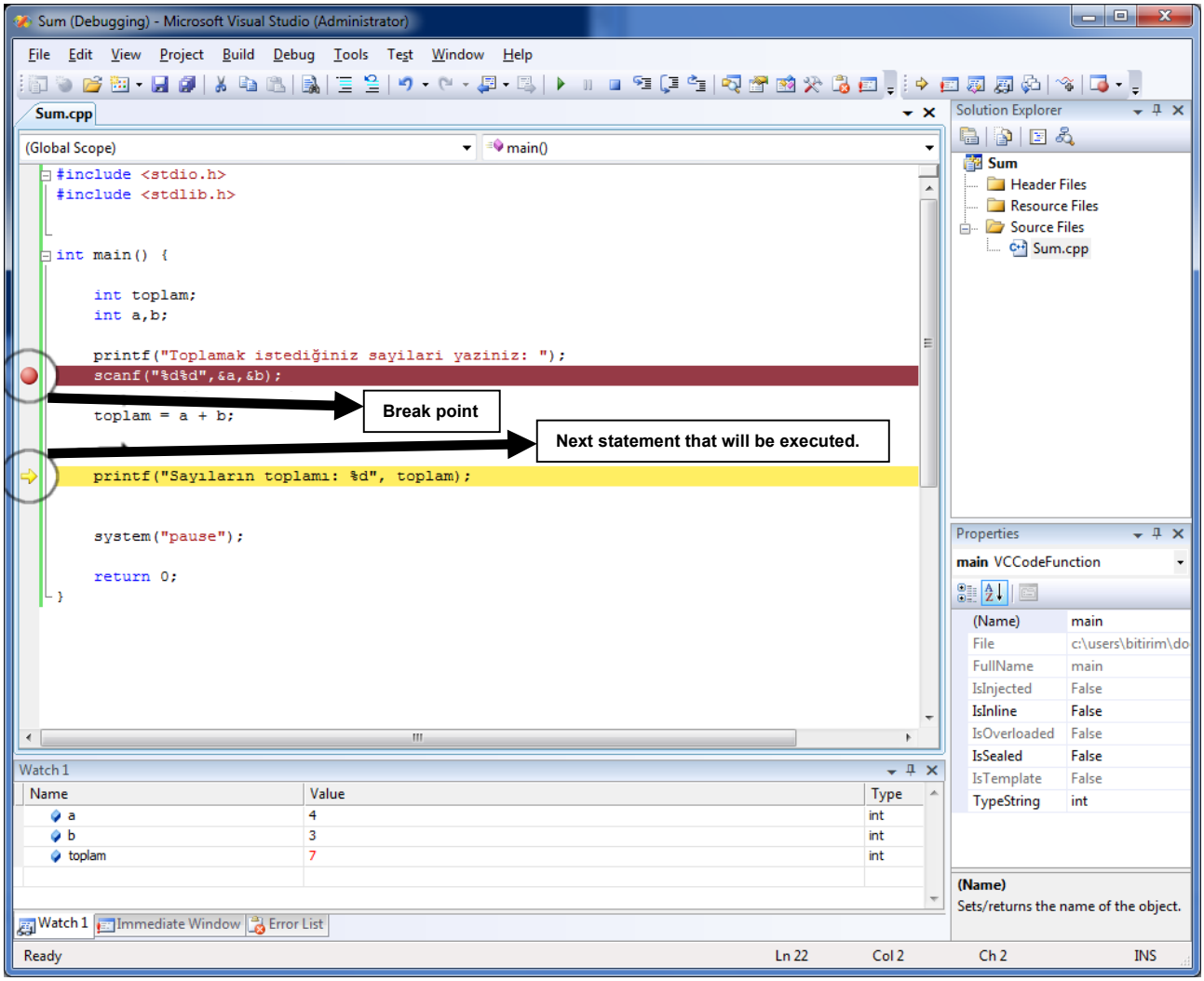

Figure 1: *Watch window*

After using the methods described above to pause your program, you may “Step”, “Continue” or “Stop” your program. To run your program step by step use “Step over” (F10) command. You can use F5 or run button to continue the execution of your program. If there are break points afterward, your program will pause again. To terminate your program use “Stop debugging” (Shift + F5) command located in “Debug” menu.

While your program is paused, you have the chance to see the contents of your variables. Right click and use “Add watch” command to see the value of a variable. This command will place your variable

to the "Watch" window and its result will be displayed next to it. In Figure 1 a watch window containing three variables is shown as well as break point and current execution line. If the watch window is not visible, you can open it using Alt + F3.

Experiment

1. Write a program that produces the following output using **printf** function. Note: you only need to use asterisk (*) and space characters.

```
*
**
***
****
 ***
  **
   *
```

2. Ask length and height of a rectangle from the user, calculate the area and display it on the screen.

3. Find the errors in the given program (should be available on the web).

4. Read two points (two x and y values; x1, x2, y1, y2). Using these values calculate and display the line that passes through those points in the following form.

```
y = Ax + B
```

You may use following formula to calculate A and B values.

$$A = \frac{y2 - y1}{x2 - x1}$$

$$B = y2 - x2 * A$$

Sample output:

```
y = 5x + 4
```

5. Write a program that calculates the position of a vehicle at 10^{th}, 20^{th} and 30^{th} seconds. Read the speed vector (containing x and y values) and initial position of the vehicle from the user. Assume that the given speed is in m/s and the position is measured in meters. You can use the following formula

$$x_{end} = x_{start} + v * t$$

Note: While coding your program, try to provide useful information about how to use the program or what is the displayed value. Otherwise, your program will not be usable by anyone else. Even you

might have trouble using it later on. Also, writing comments will help you or the person who needs to read or modify your code.

6. Write the following program and run it step by step and follow the values of a, b and sum variables.

```
#include <stdio.h>
#include <stdlib.h>


int main() {

   int sum;
   int a,b;

   printf("Type two numbers you wish to sum: ");
   scanf("%d%d",&a,&b);

   sum = a + b;


   printf("The sum of the numbers is %d", sum);


   system("pause");

   return 0;
}
```

7. After writing previous program, place breaks points on several lines and explore your opportunities.

8. Run the 5th program step by step. Also check the values of your variables.