BSc (Hons) in Information Technology
Year 1

Lab Sheet 02

IT1010 – Introduction to Programming Semester 1, 2022

Objectives:
At the end of the class the students should be able to:

- Debug and find logical errors in the C program using Dev C++ IDE

Debugging is the process of detecting and removing potential logical errors (also called as 'bugs') in a computer program.

Exercise 1

1. You are supposed to write a C program to input marks earned for the exam (out of 100) and the marks earned for the homework (out of 100) to calculate and display the overall course score.

   Students can obtain only fifty percent from their exam and from their homework.

   Score 1 = exam score * (50 / 100.0)
   Score 2 = homework score * (50 / 100.0)

   The overall course score can be calculated as follows.

   overall score = score 1 + score 2

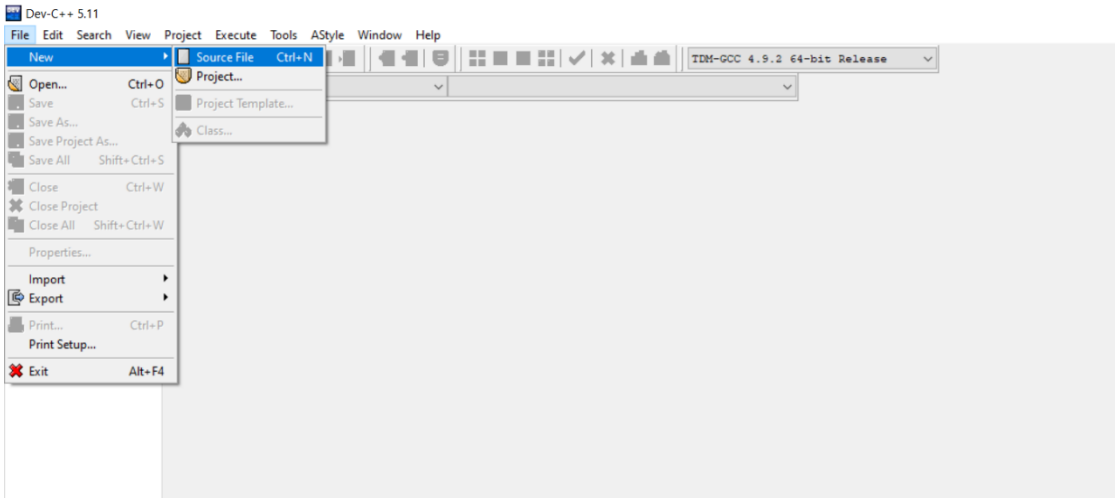
2. Follow the following steps to write a sample program for the above problem and save it as **excercise1.c**

   Step 01
   First, create a folder in your desktop and name it as **Lab2**.

Step 02

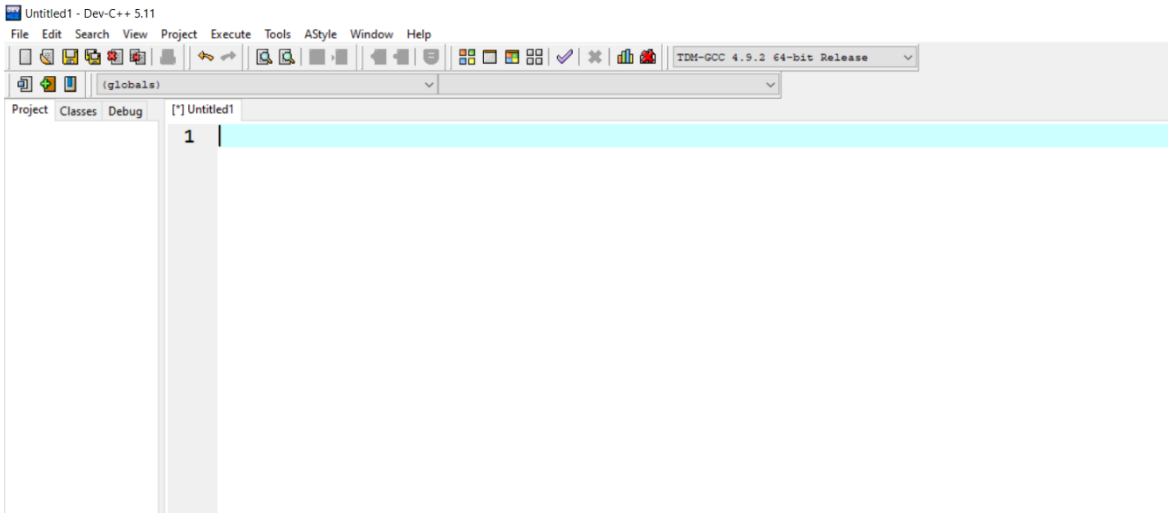
Then, open Dev C++ IDE and create a source file.

File → New → Source File



Step 03

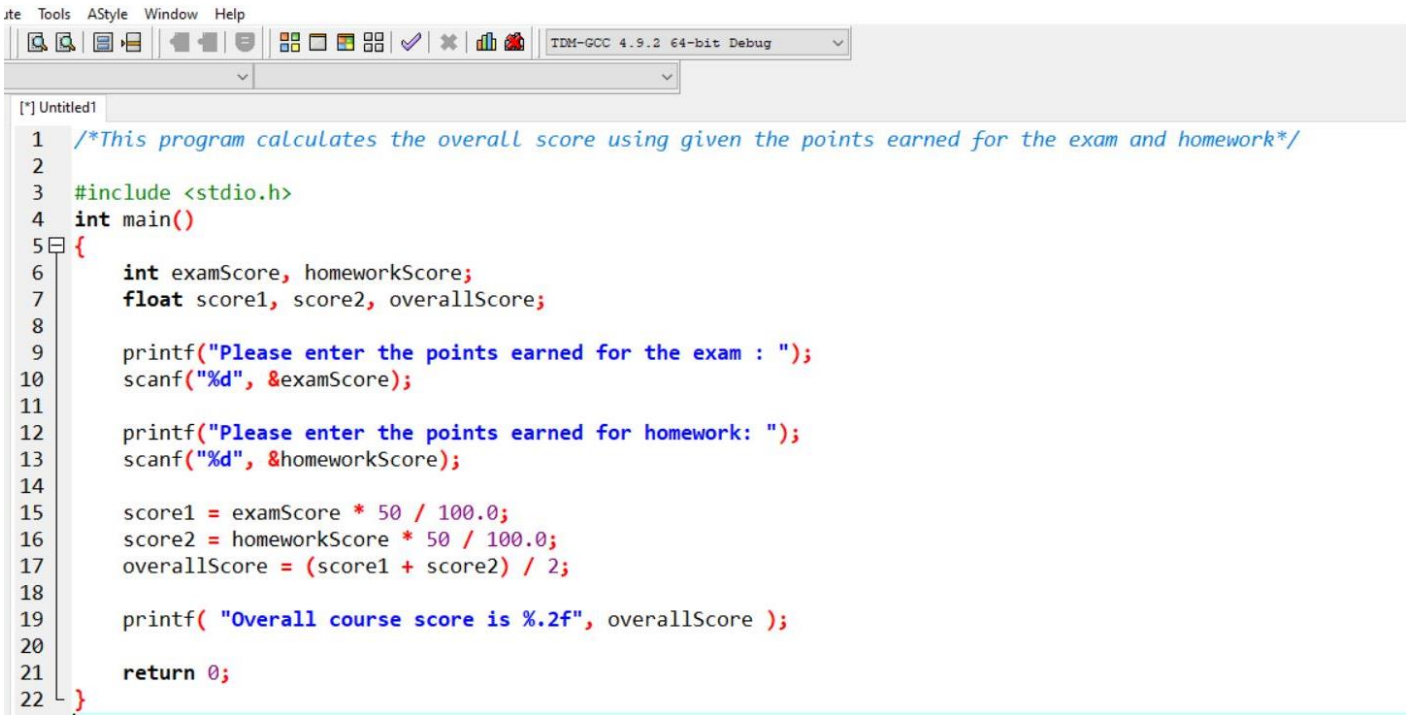
A source file will be created as below.



Step 04

Now, type the following C program.

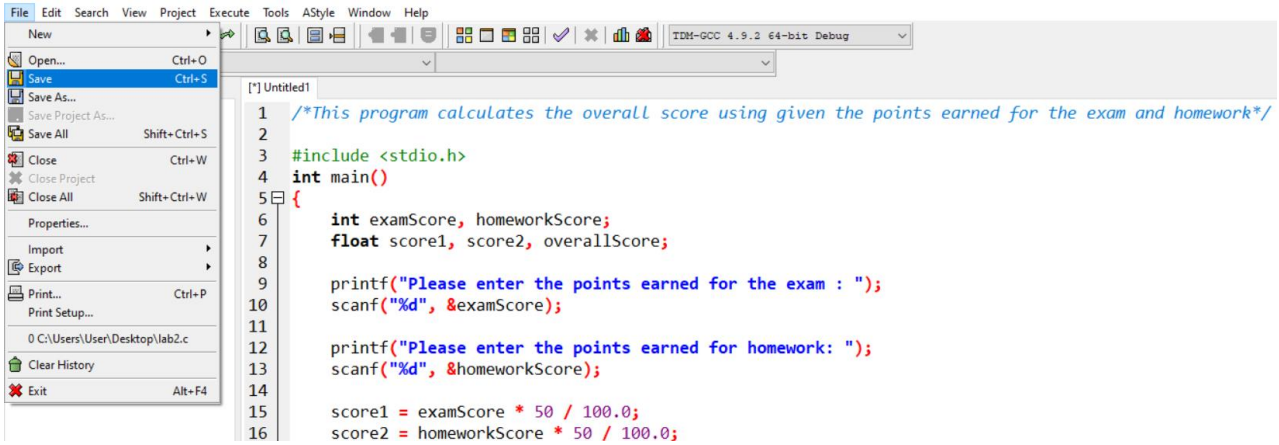
```
/*This program calculates the overall score using given the points
earned for the exam and homework*/

#include <stdio.h>
int main()
{
      int examScore, homeworkScore;
      float score1, score2, overallScore;

      printf("Please enter the points earned for the exam : ");
      scanf("%d", &examScore);

      printf("Please enter the points earned for homework: ");
      scanf("%d", &homeworkScore);

      score1 = examScore * 50 / 100.0;
      score2 = homeworkScore * 50 / 100.0;
      overallScore = (score1 + score2) / 2;

      printf( "Overall course score is %.2f", overallScore );

      return 0;
}
```

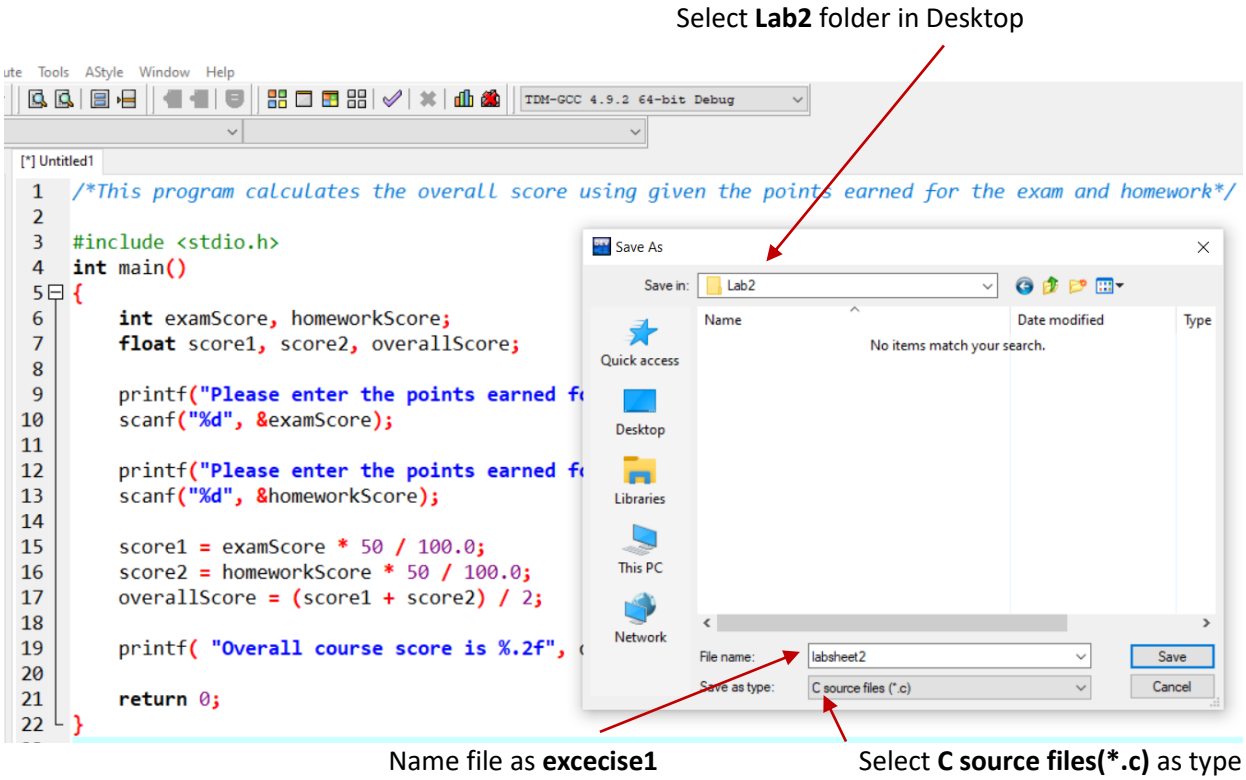


Step 05

Save the source file as **excercise1** inside the folder called **Lab2** in your desktop.

File ⟶ Save





Here, you have saved your source file as a C file called **excercise1.c**

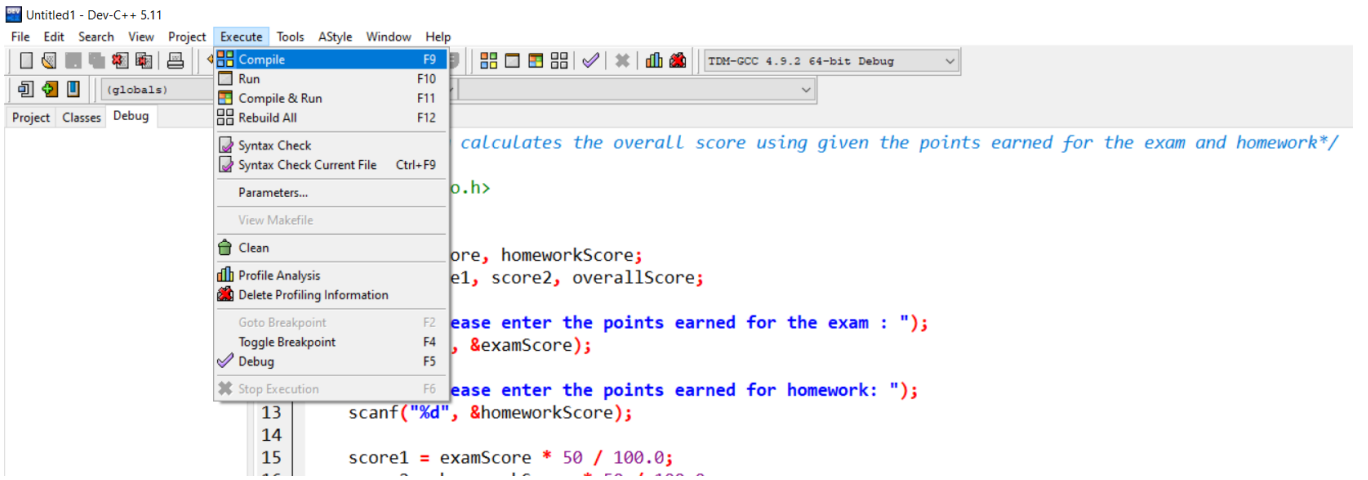
Step 06

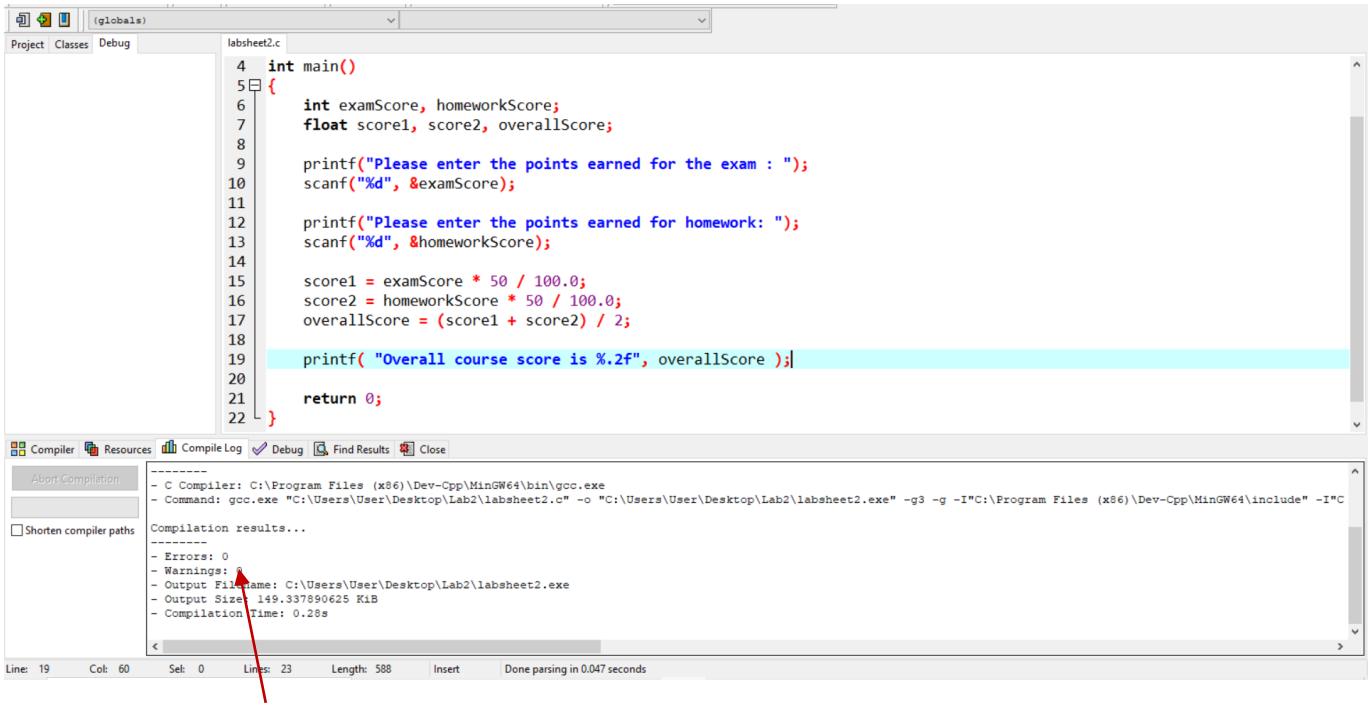
Compile C file.

Execute → Compile

or

Shortcut key : F9



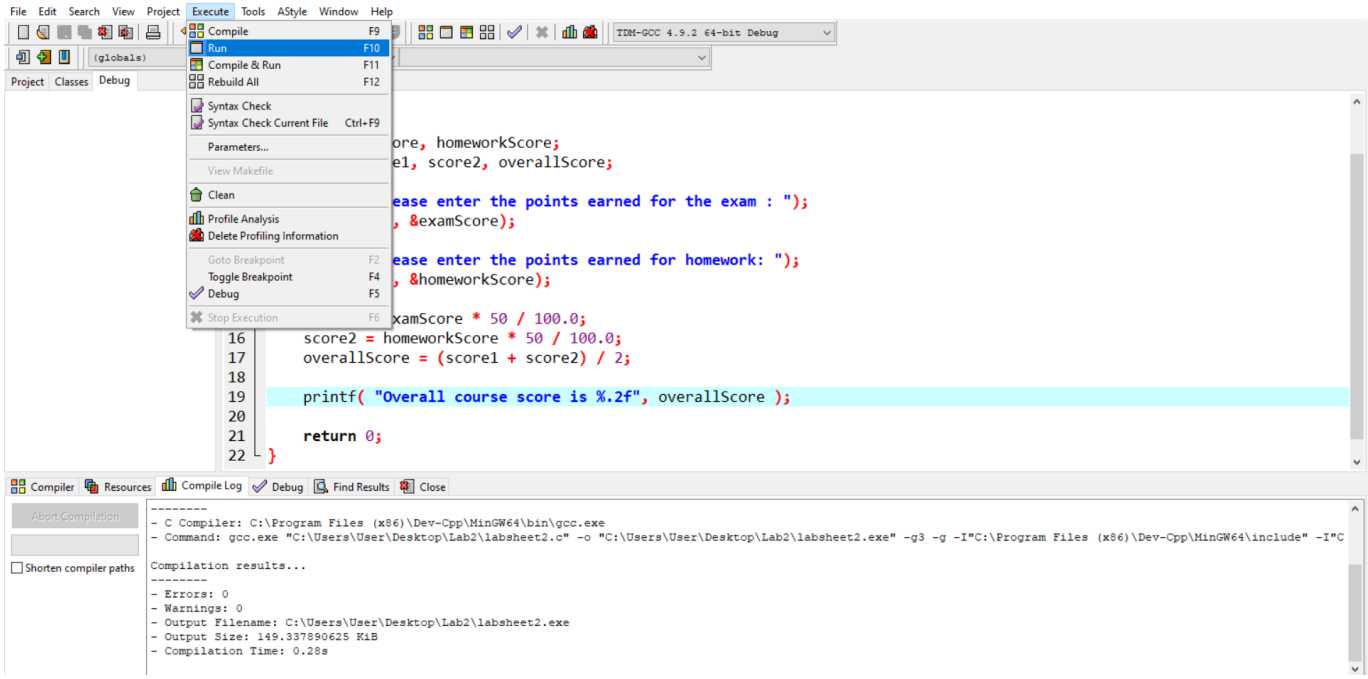


Observe the compilation errors and warnings

Step 07

If your program does not have any errors and warnings, execute the C program.

Execute ⟶ Run
or
Shortcut key : F10



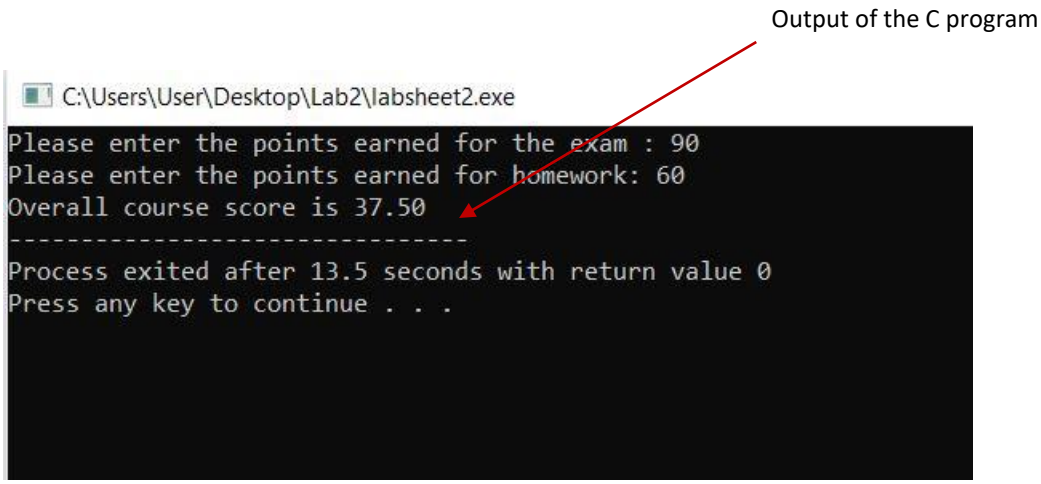
Step 08

Run your program with the following sample data set.

The points earned for the exam: 90
The points earned for homework: 60

If you have calculated overall course score manually, the expected overall score should be 75.

Now, compare the expected overall score with the program output.



Here, you can see that the expected value and the program output is different. It means, there is/are logical error/s in the given C program.

3. Now, you can use the debugging option to find logical errors in the program.

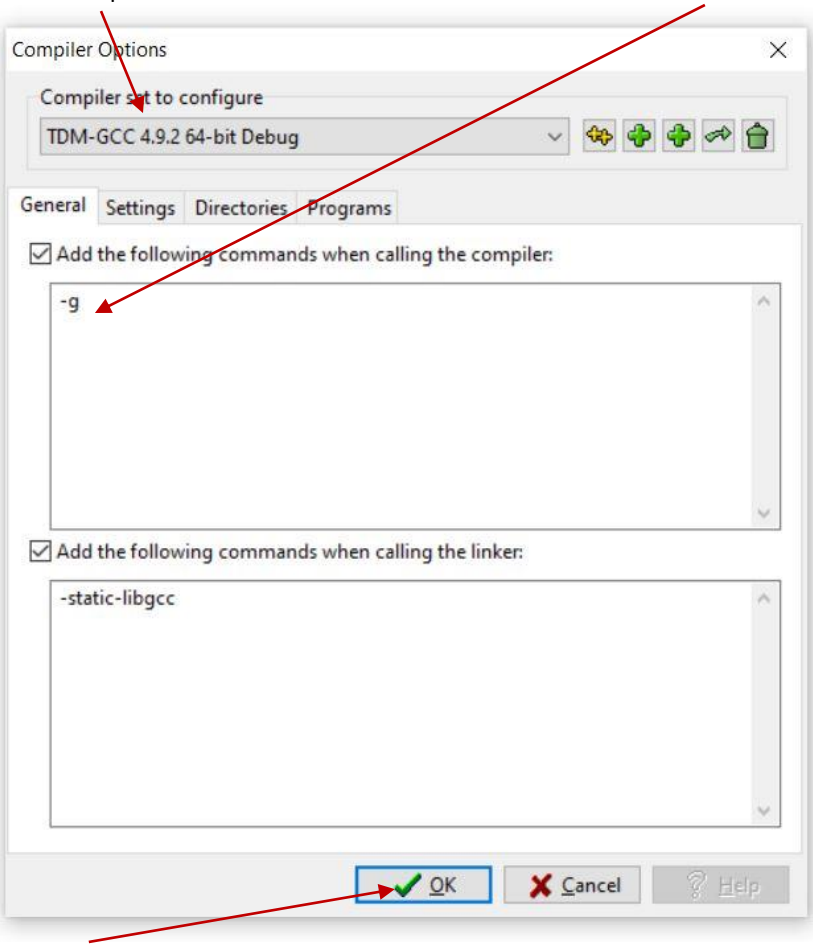
Step 01

You need do some setting changes in Dev C++ IDE

Go to Tools → Compiler options

Select this option

Add a tick here and type **-g**



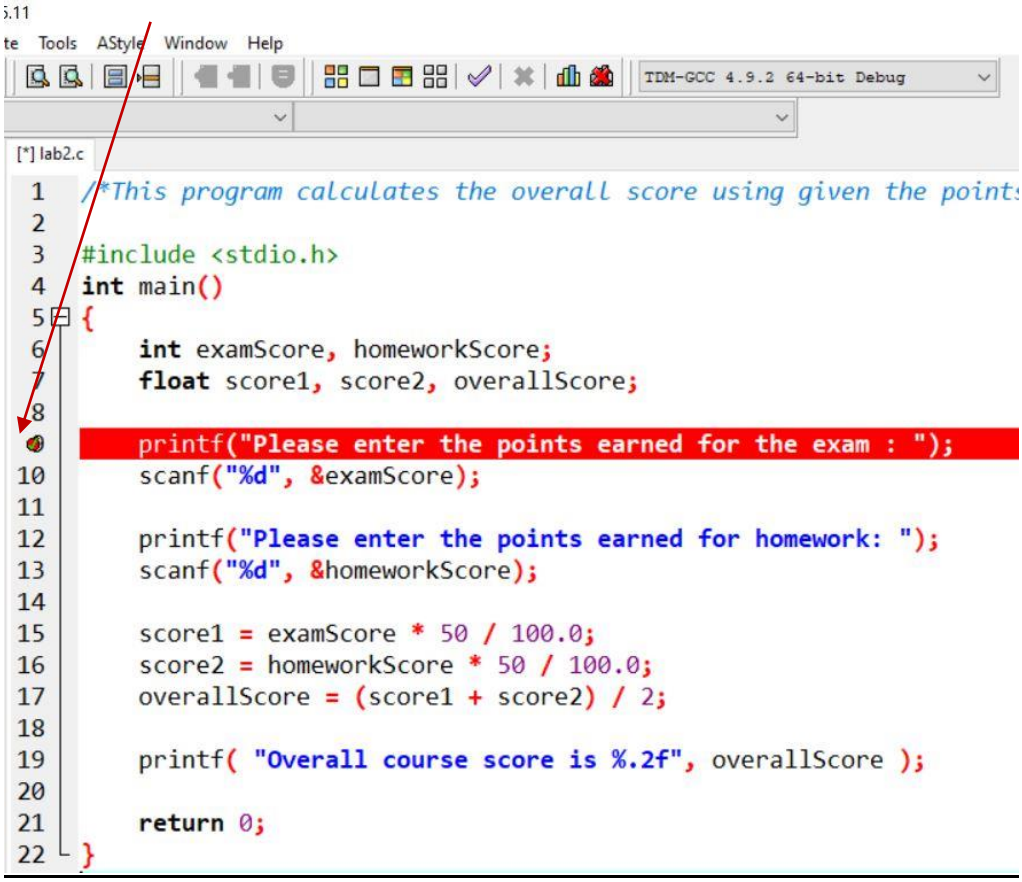
Press **OK** button

Step 02

Set break points in C program

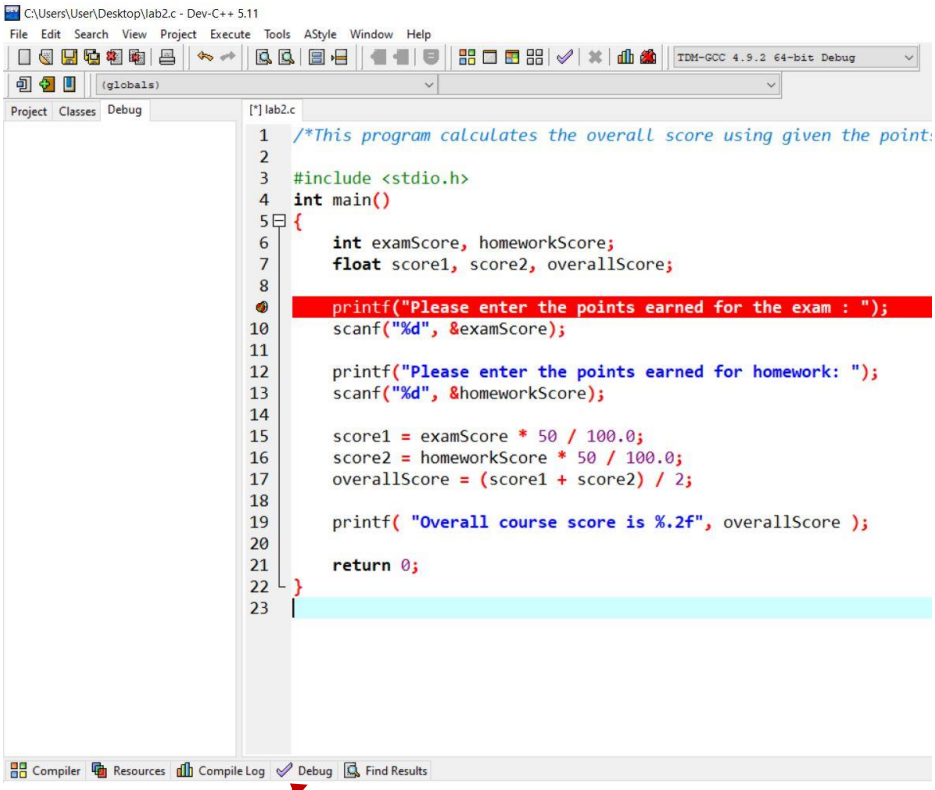
A breakpoint is a point in the program where you want the execution to stop temporarily so that you can examine the values of variables.

To set a break point, click on the line number of relevant statement. Here, a break point is set at **line number 9**.



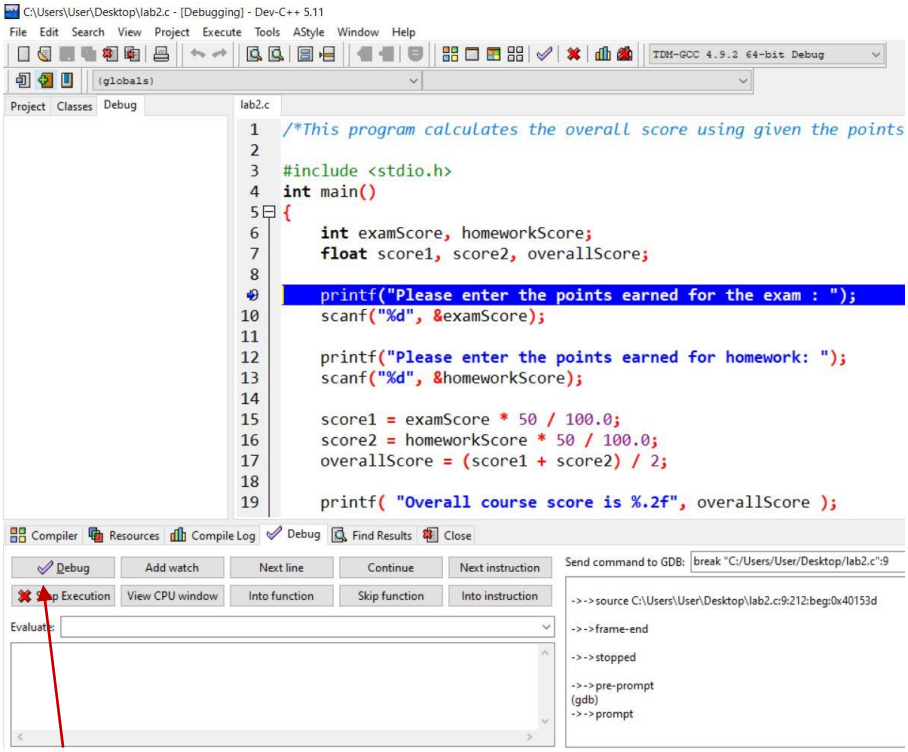
Step 03

Start debugging



Click on the **Debug** button

Then, debug window will appear as follows.

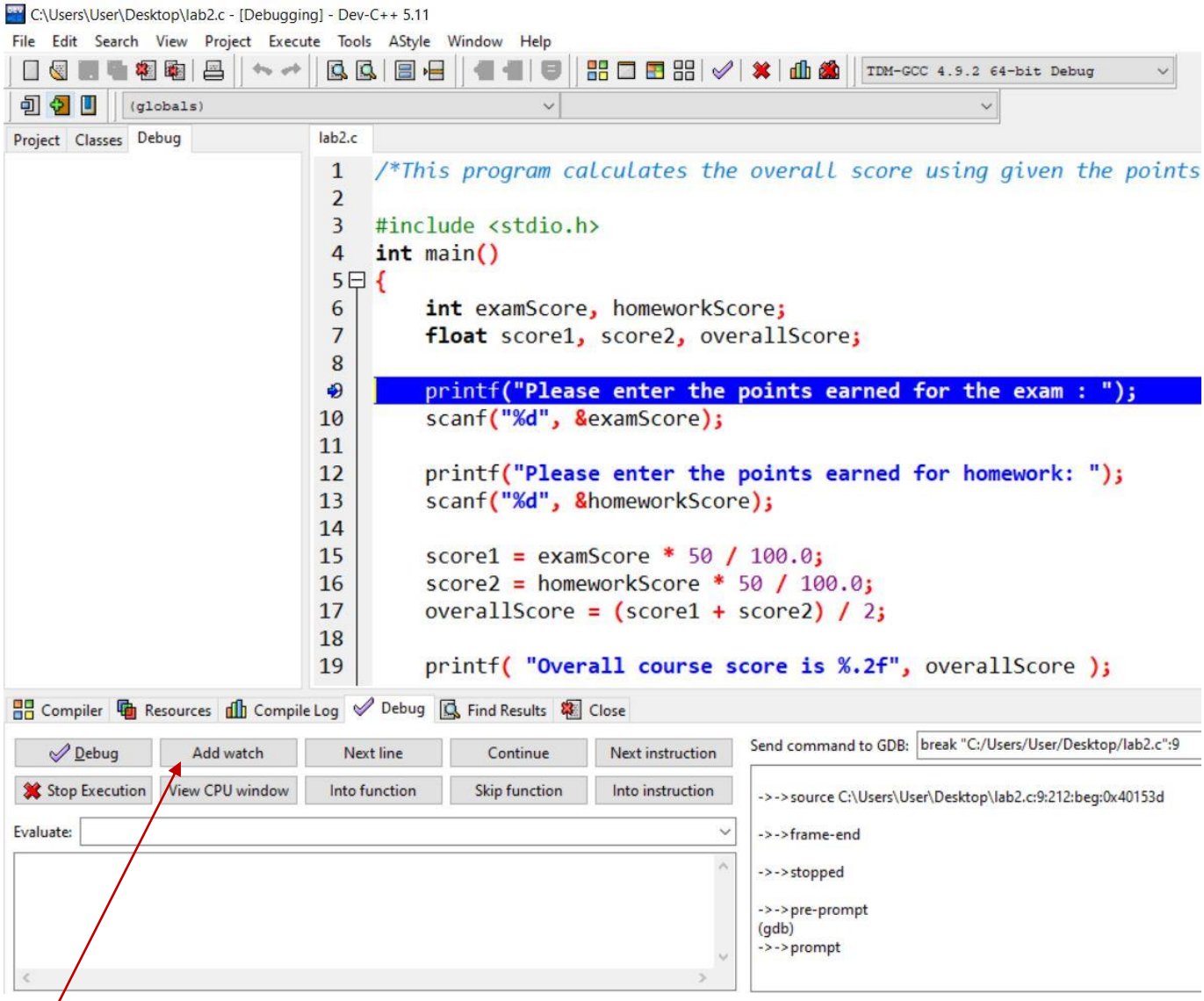


Click on **Debug** button, then your program will be executed up to **line no. 8**

The examScore, homeworkScore, score1, score2 and overallScore variables are declared and not initialized, but you can check what are the values (garbage values) that are stored in these memory locations.

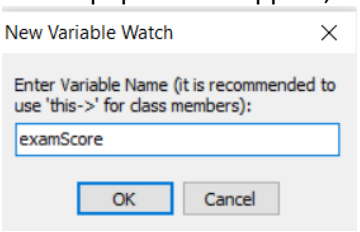
Step 04

Add a watch on a variable



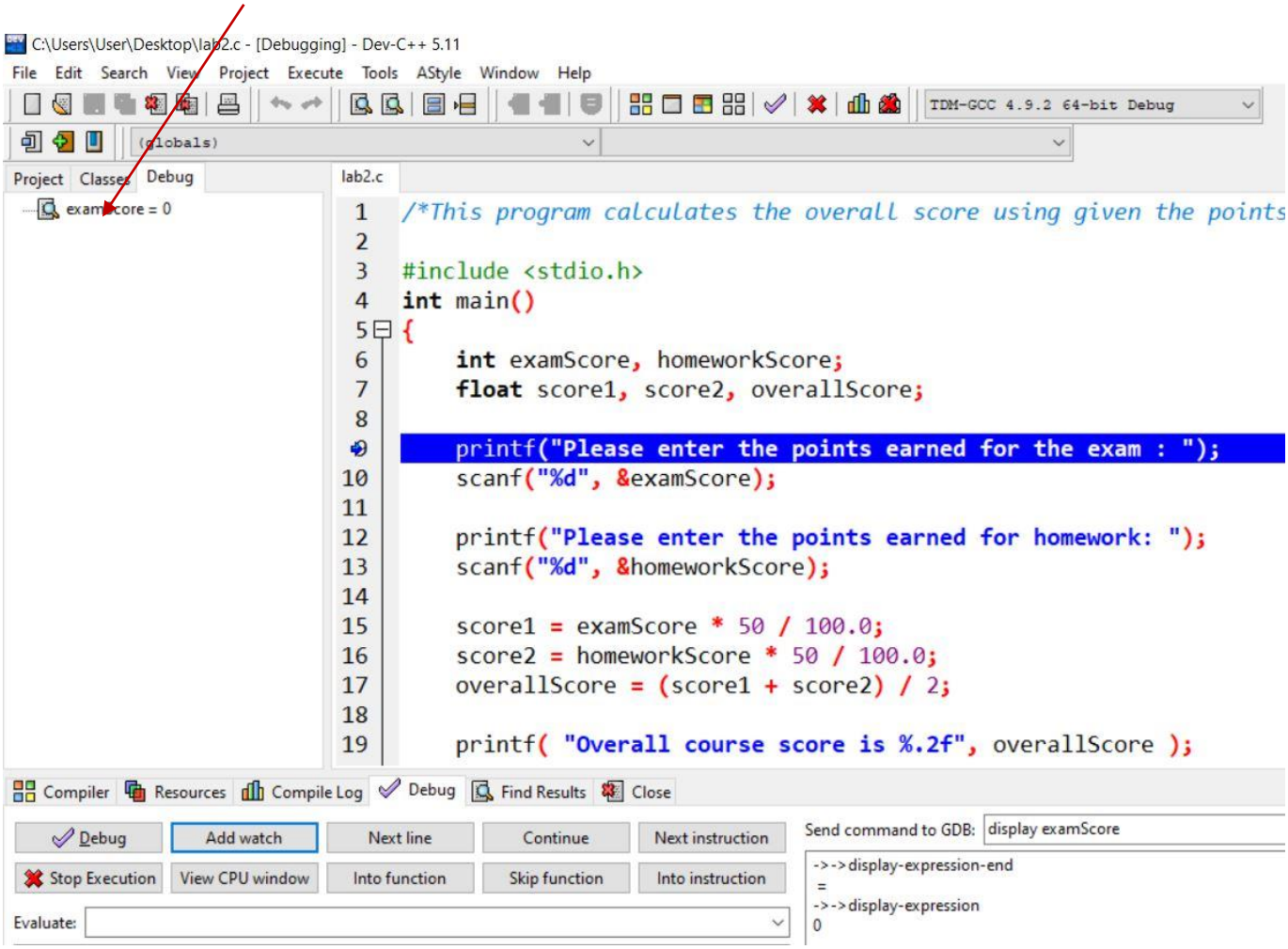
Click on **Add watch** button

Then a pop box will appear,



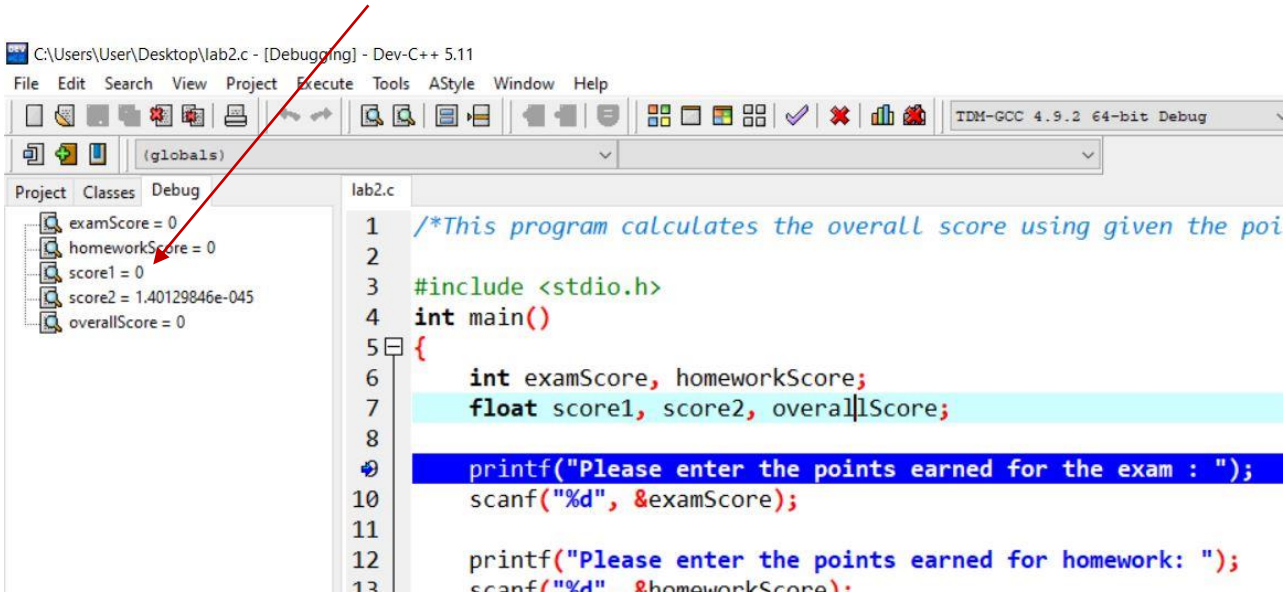
Here, you can give a name of a variable that you are going to add a watch, then click on **OK** button. Now, a watch will be added to examScore variable.

Although we haven't store any value in this variable, by default **zero** is stored in examScore variable.



Step 05

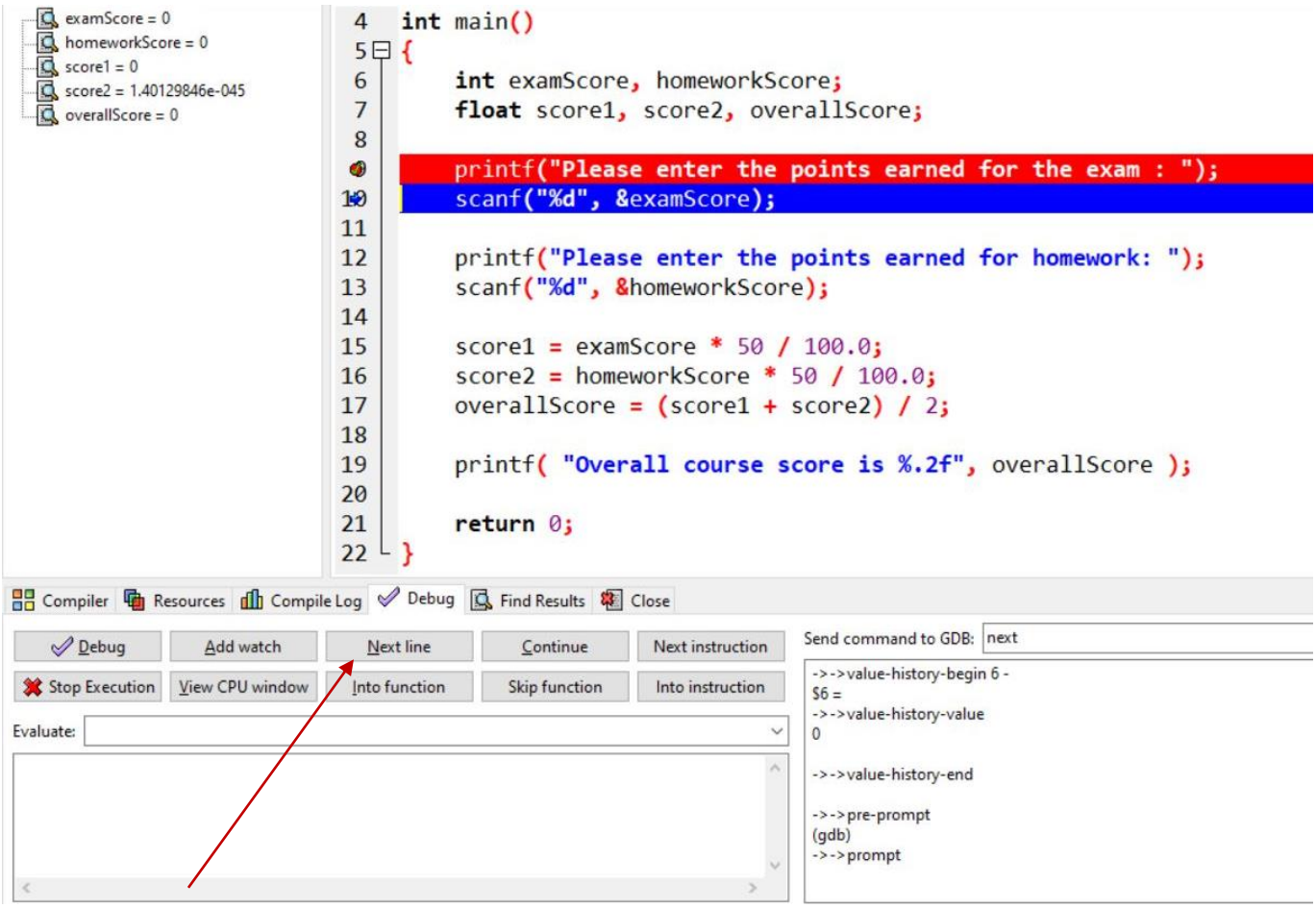
Add watches on other variables and check their values. (homeworkScore, score1, score2, overallScore)



Now, the program is executed up to **line no. 8**.

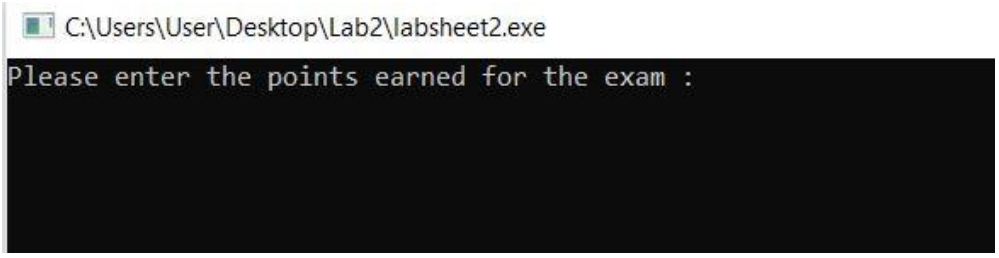
Step 06

To execute next C statement in line number 9, click on **Next line** button.



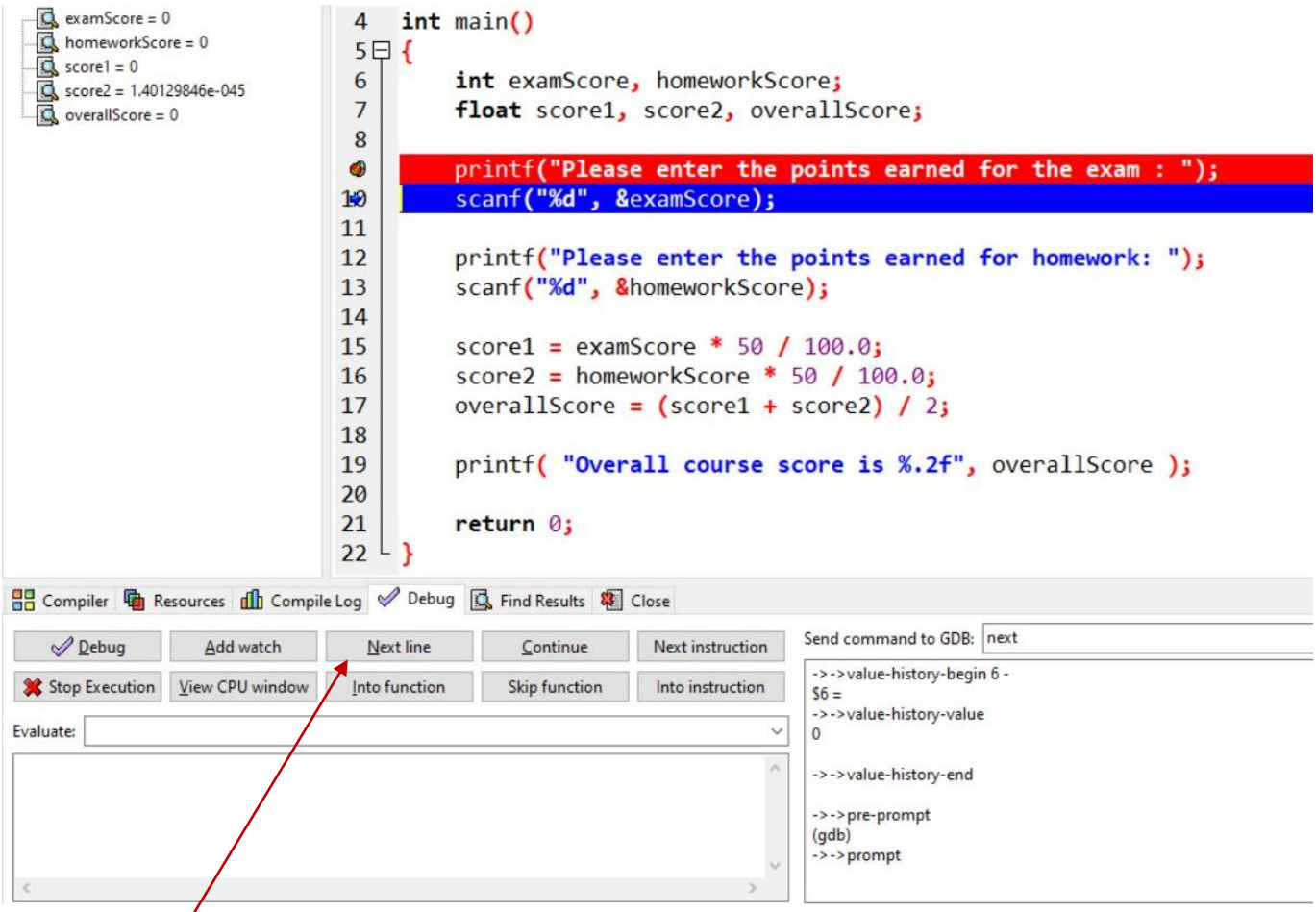
Click on **Next line** button

Then, **line number 9** will be executed. Now you can see the output window as follows.



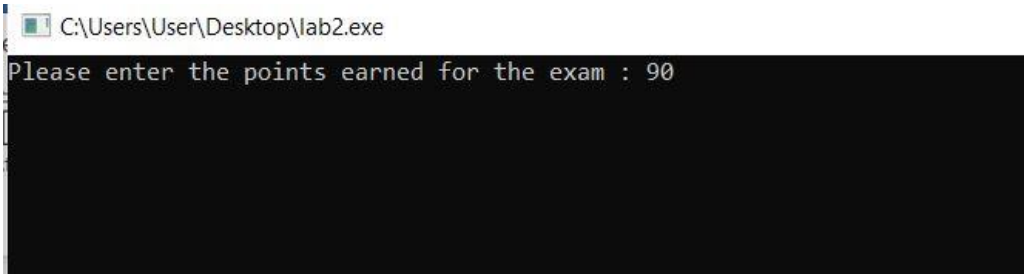
Step 07

To execute next statement line in number 10, click on **Next line** button again.



Click on **Next line** button

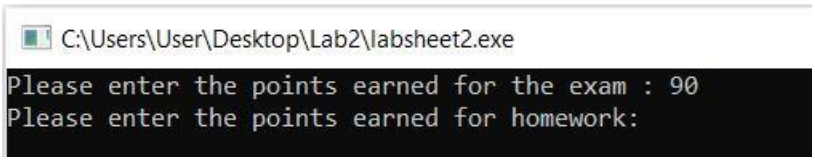
Then, **line no. 10** will be executed. Now you can input the points for the exam as 90. Then, press **enter** button in your keyboard.



Step 08

To execute next statement in line no. 12, click on **Next line** button again.

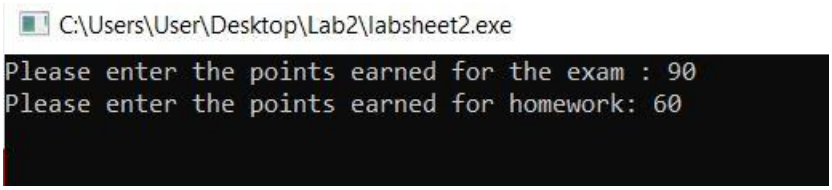
Then, **line no. 12** will be executed. Now, you can see the output window as follows.



Step 09

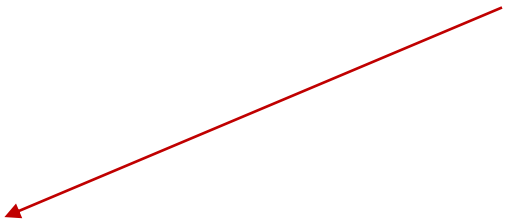
To execute next statement in line no. 13, click on **Next line** button again.

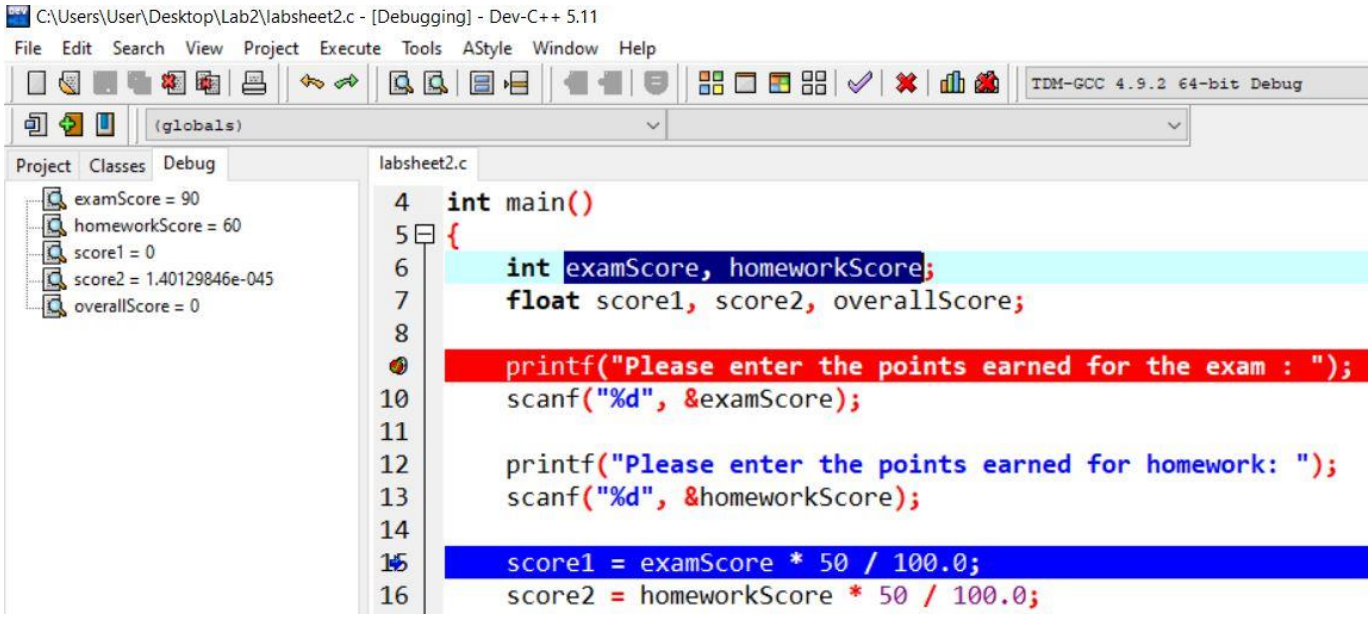
Then, **line no. 13** will be executed. . Now you can input the points for the homework as 60. Then, press **enter** button in your keyboard.



Step 10

Now, you can see that the variable values of examScore and homeworkScore are changed to 90 and 60.



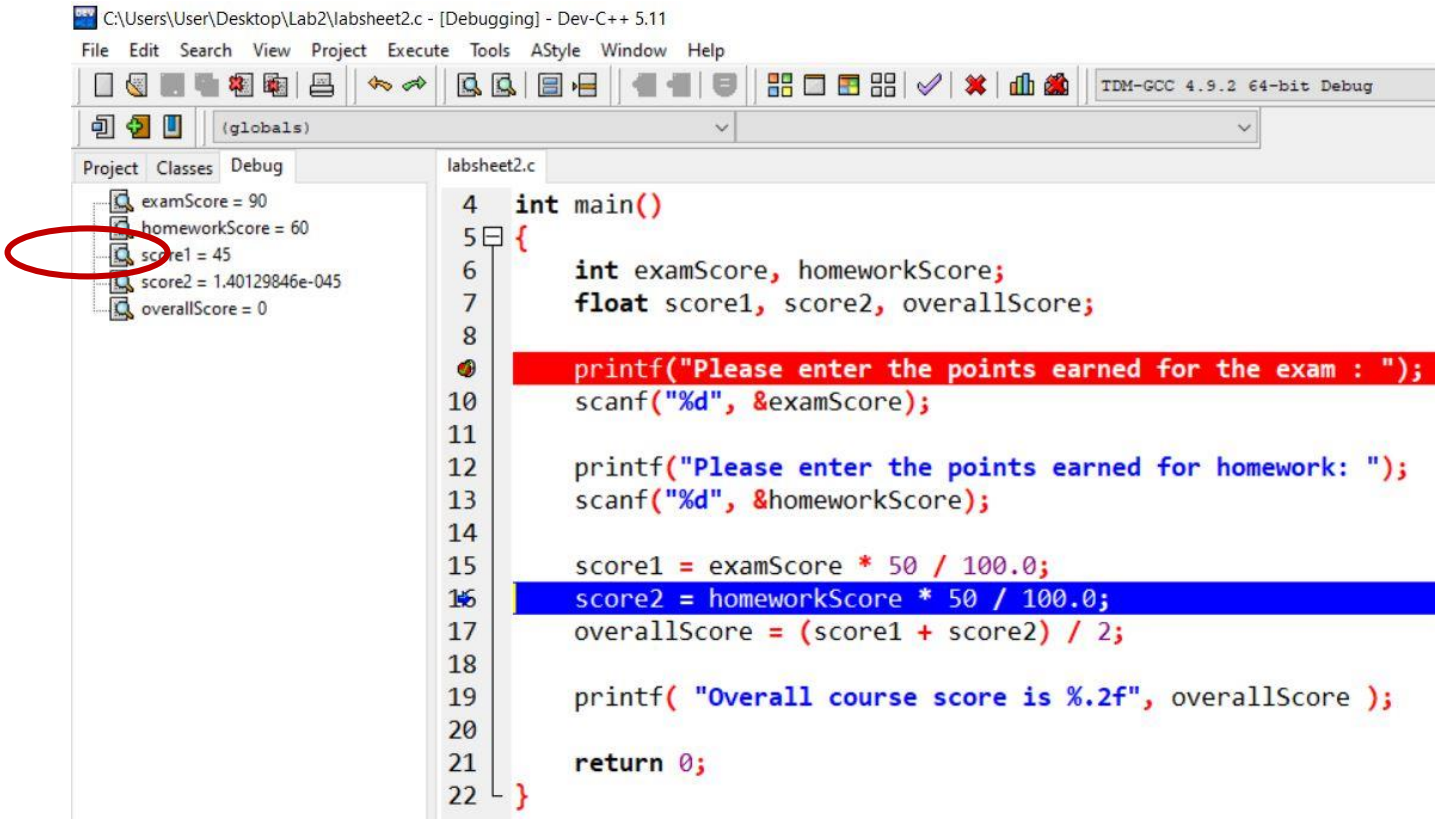


Step 11

To execute next statement in line no. 15, click on **Next line** button again.

Then, **line no. 15** will be executed and **score1** value is calculated.

45 is stored in **score1** variable. It means that this calculation is correct, and it takes 50% of exam marks. There is no logical error in this calculation.

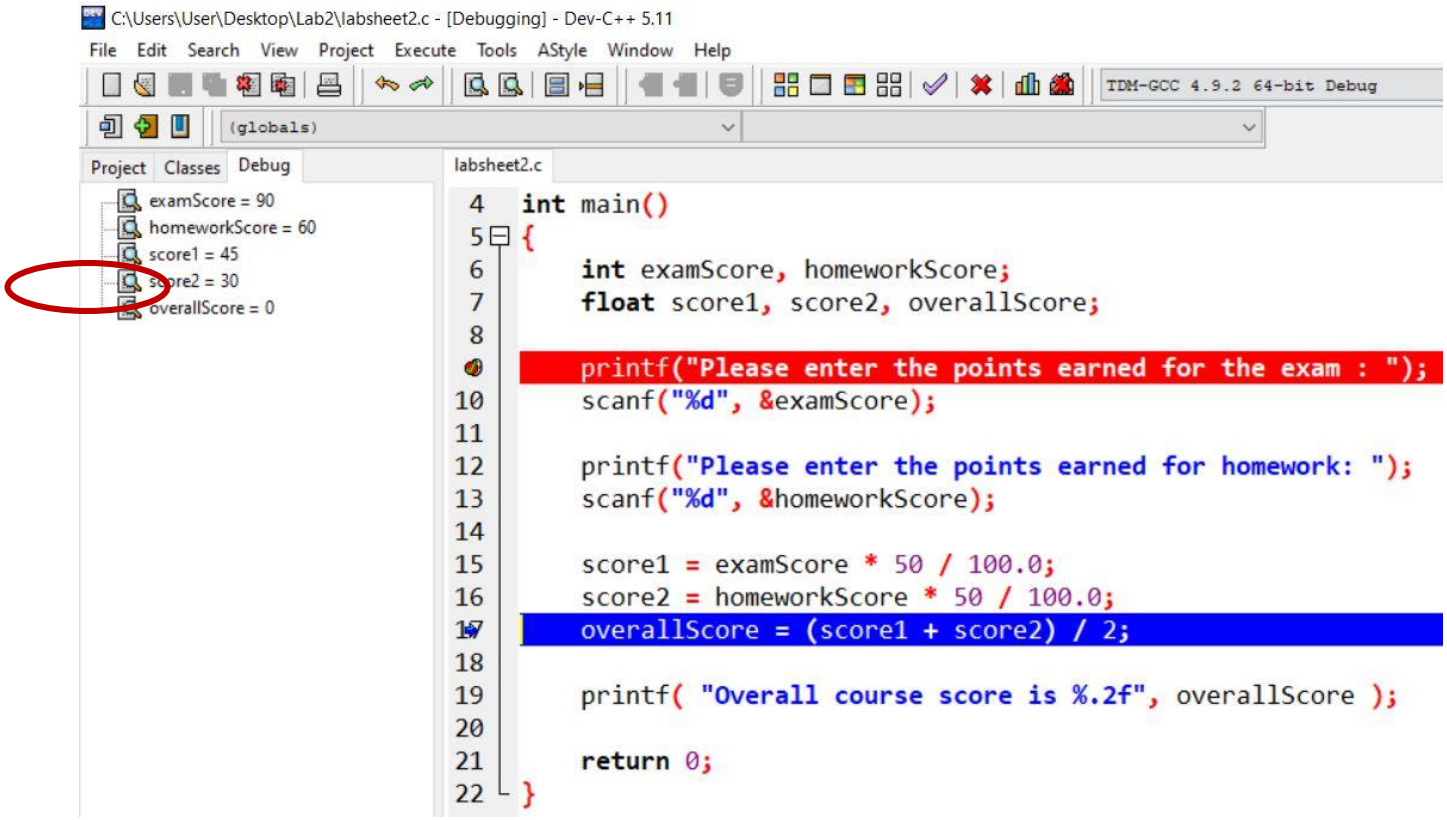


Step 12

To execute next statement in line no. 16, click on **Next line** button again.

Then, **line no. 16** will be executed and **score2** value is calculated.

30 is stored inside **score2** variable. It means that this calculation is correct, and it takes 50% of exam marks. There is no logical error in this calculation.

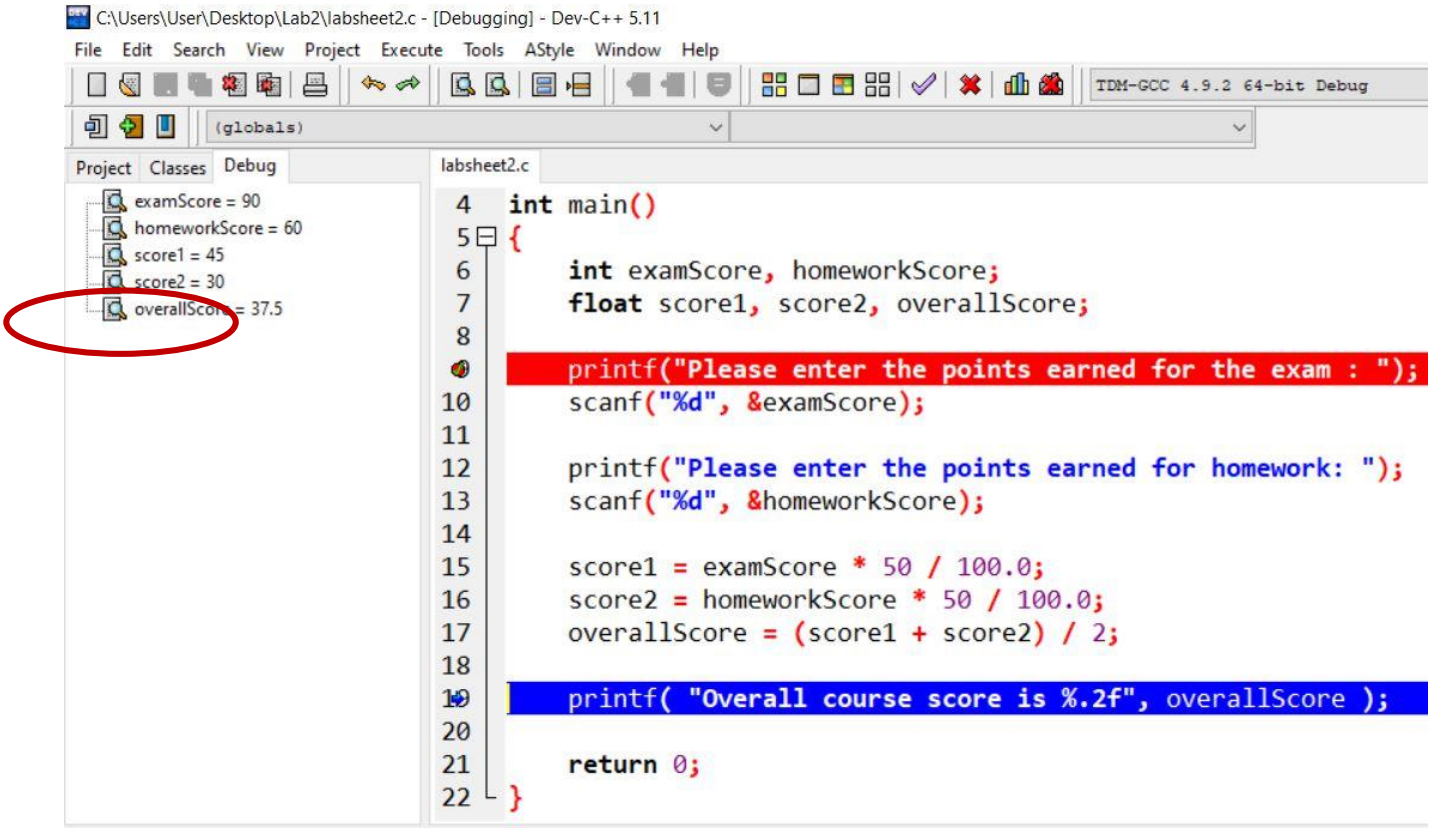


Step 13

To execute next statement in line no. 17, click on **Next line** button again.

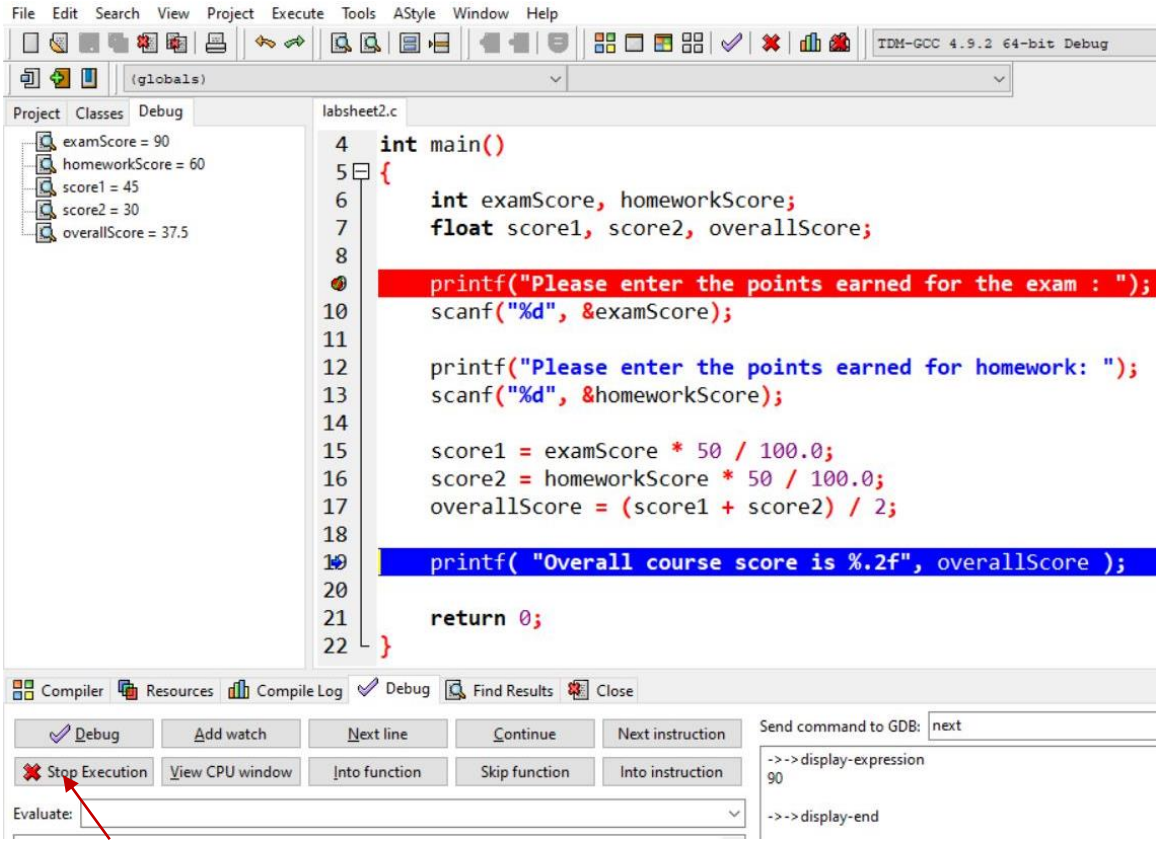
Then, **line no. 17** will be executed and **overallScore** value is calculated.

37.5 is stored in **overallScore** variable. You can see that this calculation is incorrect, and it doesn't calculate overall score correctly.
So that, there can be a logical error in this calculation in line no. 17.



Step 14

To fix the logical error, debugging process should be stopped using **Stop Execution** button.



Click on **Stop Execution** button

Step 15

When we observe the statement in line no. 17.

```
overallScore = (score1 + score2) / 2;
```

Here, we don't need to divide the addition of two scores by 2 since we need to take overall score out of 100.
That's the error that we have done.
Now, you can modify the statement as bellow.

```c
int main()
{
    int examScore, homeworkScore;
    float score1, score2, overallScore;

    printf("Please enter the points earned for the exam : ");
    scanf("%d", &examScore);

    printf("Please enter the points earned for homework: ");
    scanf("%d", &homeworkScore);

    score1 = examScore * 50 / 100.0;
    score2 = homeworkScore * 50 / 100.0;
    overallScore = score1 + score2;

    printf( "Overall course score is %.2f", overallScore );

    return 0;
}
```

Step 16

Click on the line number of the relevant statement which includes the break point to remove it.

Step 17

Compile and run the program and see whether the program works as expected.

If you can't get the expected output, there may be more logical errors.
Then you need to debug your program again to identify those logical errors.

Hint : When you're going to debug your program again, you can set break point at line no. 17 since in earlier process, we have confirmed that up to line no. 16, there are no logical errors.

Exercise 2

Write a C program to calculate the summation of two integer values that input from the keyboard.

Step 01
Identify the inputs, calculations and outputs of the given problem.

Step 02
Create a source file using Dev C++ IDE.

Step 03
Save your program as **exercise2.c** in the folder called **Lab2** in your Desktop.

Step 04
Compile and run the program to see the output.

Step 05
Set a suitable break point in your program.
Hint : A break point can be added at a statement after the variable declaration.

Step 06
Start debugging using debug buttons.

Step 07
Set watches on the variables and check their values.

Step 09
Execute remaining lines using **Next line** button and observe the variable values and output window until the program executes completely.
Through this process, the logical errors can be identified. If there are any logical errors, you can stop debugging and fix the error.