BSc (Hons) in Information Technology
Year 1

Lab Sheet 03

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

Exercise 1

1. In a festive season, a supermarket is giving 15% discount for the total payment of their customers. This is a sample C program that calculates the discount and the final payment of a customer.

   The given program includes some logical errors.

   Using debugging options in Dev C++, identify these logical errors and fix those. Finally, you need to get correct output.

```
//This program calculate discount and final payment of a customer
#include <stdio.h>
int main(void)
{
      double totalPay, finalPay, discount;

      printf("Enter total payment of a customer : ");
      scanf("%lf", &totalPay);

      discount = totalPay * (15 / 100);

      finalPay = totalPay + discount;

      printf("Discount : %.2f\n" , discount);
      printf("Final Payment : %.2f\n" , finalPay);

      return 0;
}
```

2. Type the above sample program in Dev C++ IDE and save the program as **exercise1.c** by following the steps below.

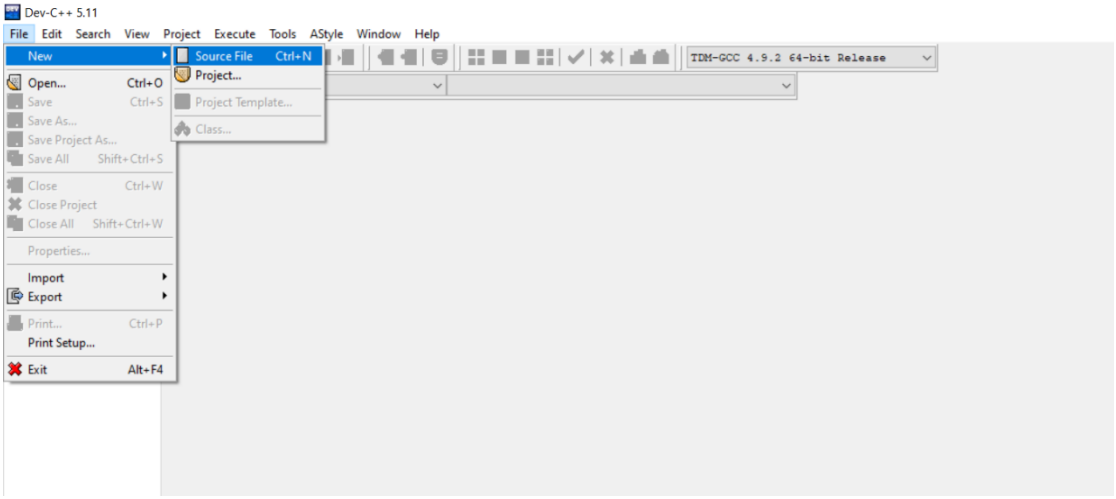
Step 01

Create a folder in your desktop and name it as **Lab3**.

Step 02

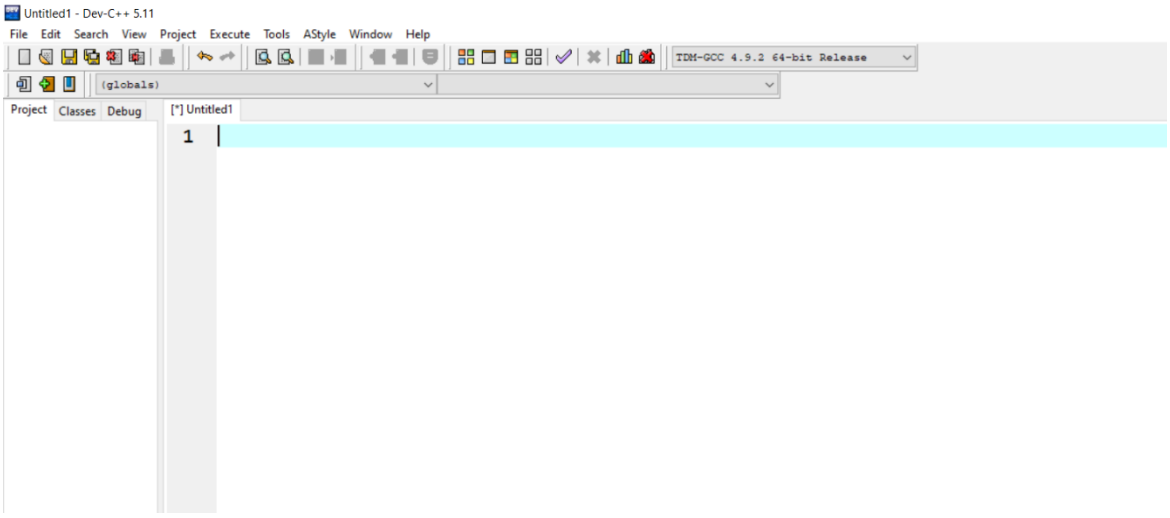
Then, open Dev C++ IDE and create a source file.

File ⟶ New ⟶ Source File



Step 03

A source file will be created as below.



Step 04

Now, type the given program.

[*] Untitled1

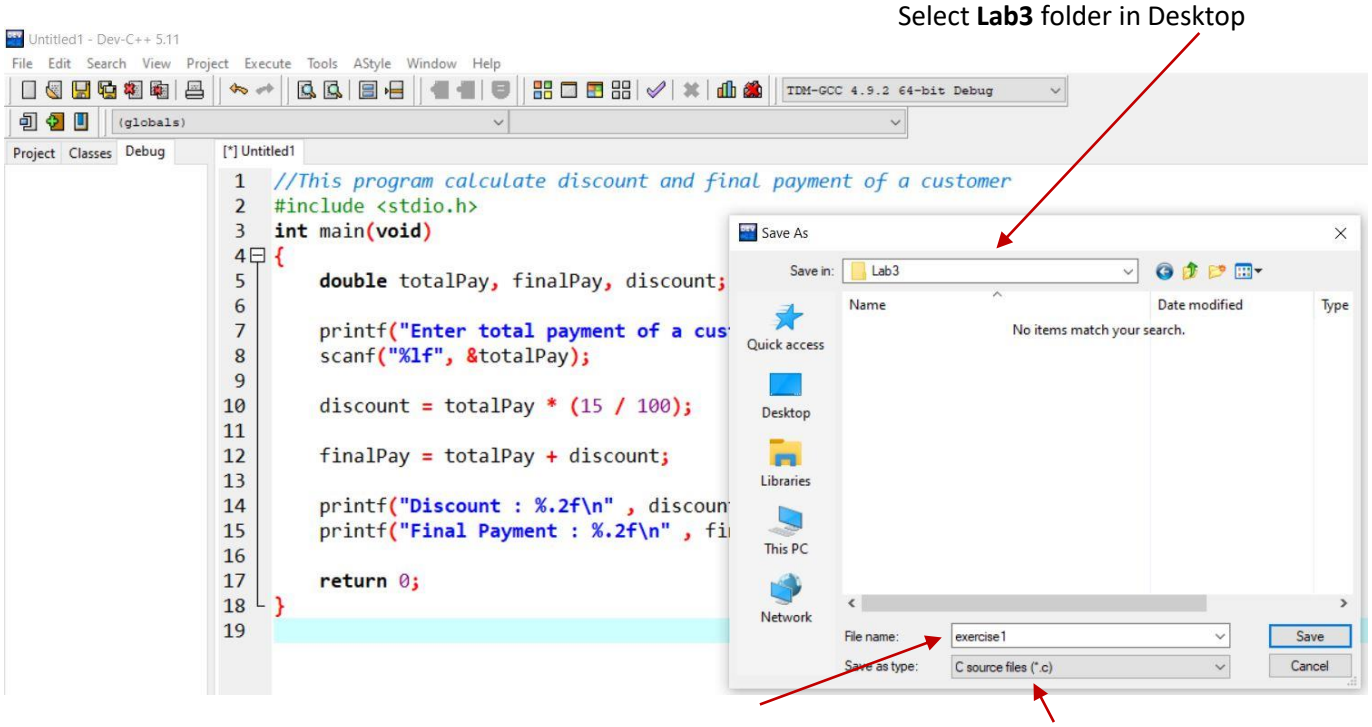
```
1  //This program calculate discount and final payment of a customer
2  #include <stdio.h>
3  int main(void)
4  {
5      double totalPay, finalPay, discount;
6
7      printf("Enter total payment of a customer : ");
8      scanf("%lf", &totalPay);
9
10     discount = totalPay * (15 / 100);
11
12     finalPay = totalPay + discount;
13
14     printf("Discount : %.2f\n" , discount);
15     printf("Final Payment : %.2f\n" , finalPay);
16
17     return 0;
18 }
```

Step 05

Save the source file as **exercise1** in the folder called **Lab3** in your desktop.

File ⟶ Save

```c
//This program calculate discount and final payment of a customer
#include <stdio.h>
int main(void)
{
    double totalPay, finalPay, discount;

    printf("Enter total payment of a customer : ");
    scanf("%lf", &totalPay);

    discount = totalPay * (15 / 100);

    finalPay = totalPay + discount;

    printf("Discount : %.2f\n" , discount);
    printf("Final Payment : %.2f\n" , finalPay);

    return 0;
}
```



Here, you have saved your source file as a C file called **exercise1.c**

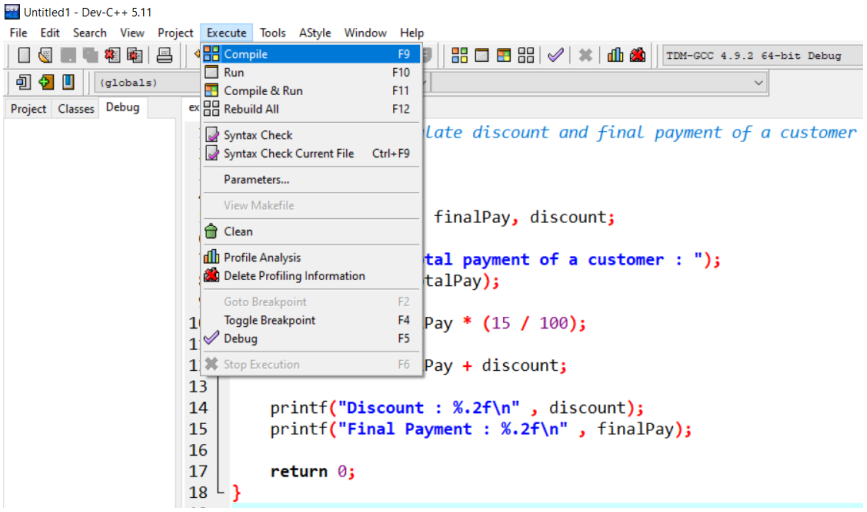
Step 06

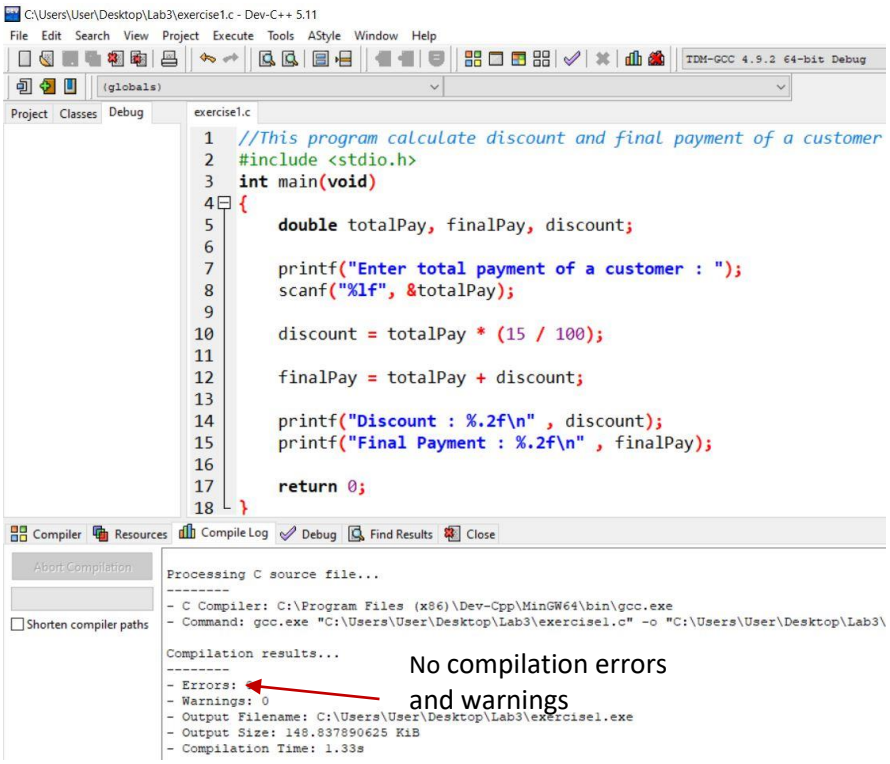
Compile C file.

Execute ⟶ Compile

or

Shortcut key : F9

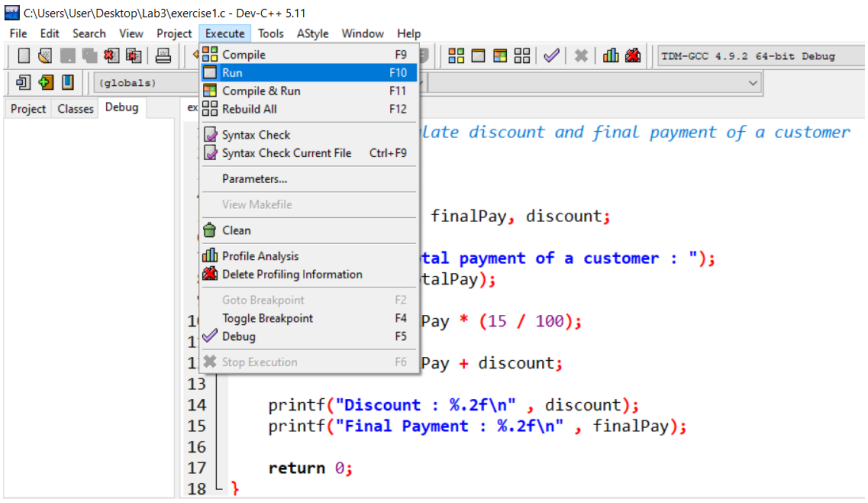




Step 07

If you don't have any errors and warnings, execute the C program.

Execute ⟶ Run
or
Shortcut key : F10



Step 08

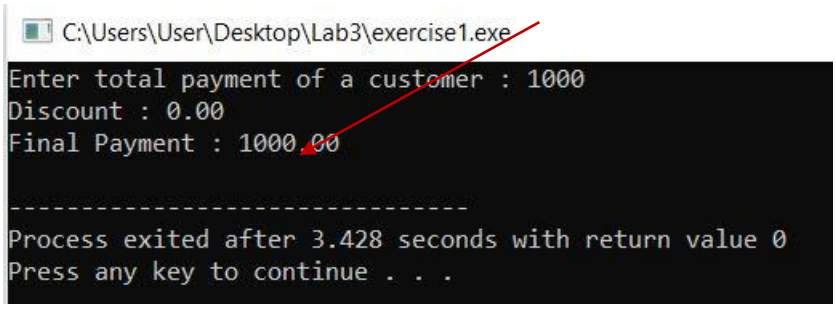
Run your program with the following sample data.

Enter total payment of a customer : 1000

If you have calculated the discount and final payment by yourselves, you will get 150 as discount value and 850 as final payment value.

Now, compare the manually calculated discount and final payment with the program output.

Output of the C program



Here, you can see that the manually calculated values and the program outputs are different. It means, there is/are logical error/s in the given C program.

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

Step 01

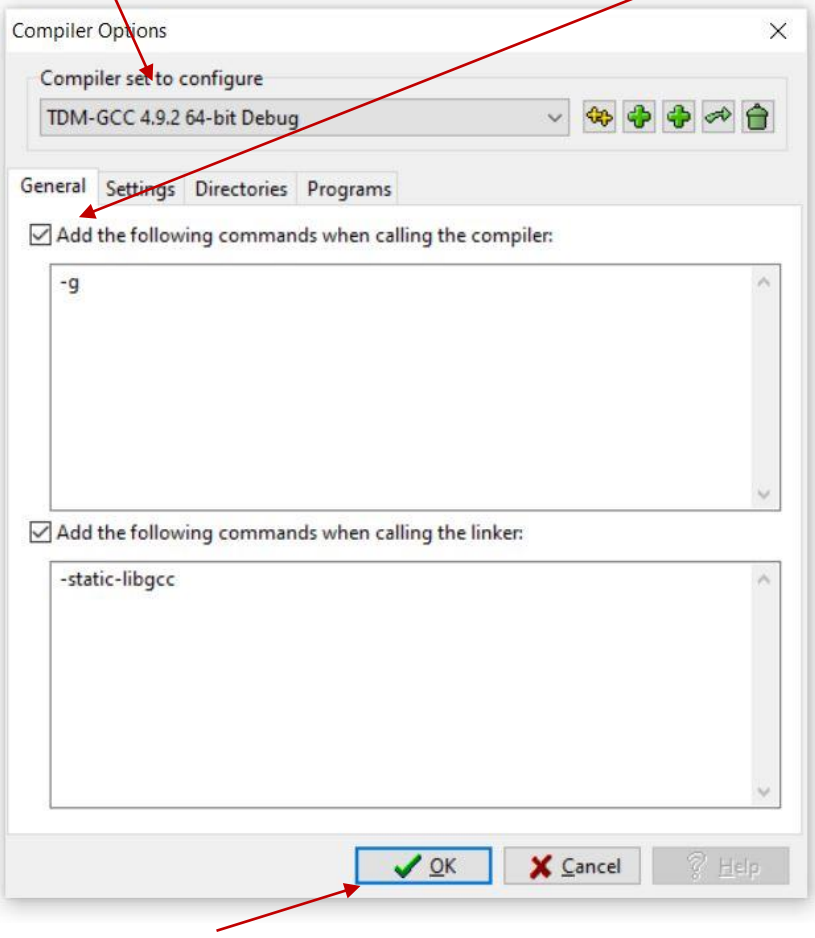
You need do some setting changes in Dev C++ IDE.

(If you have changed these settings earlier, no need to change them again.)

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.)

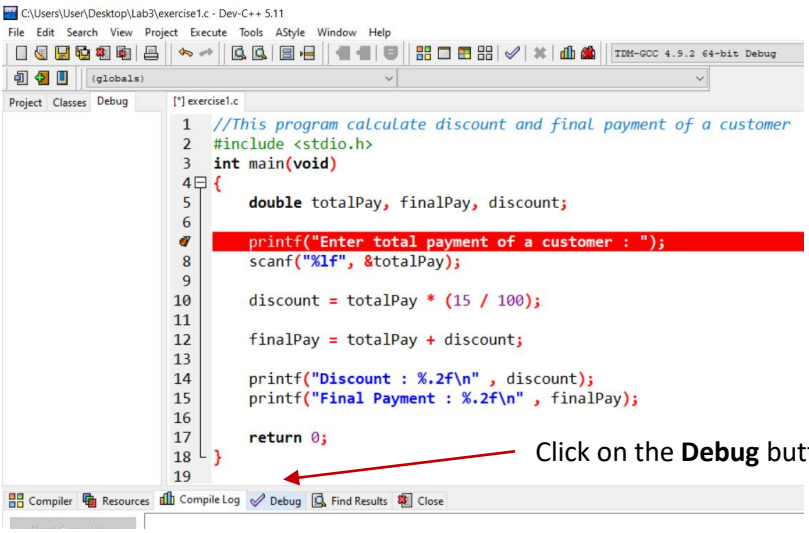
To set a break point, click on the line number of relevant statement. Here, a break point is set at **line number**

[*] exercise1.c

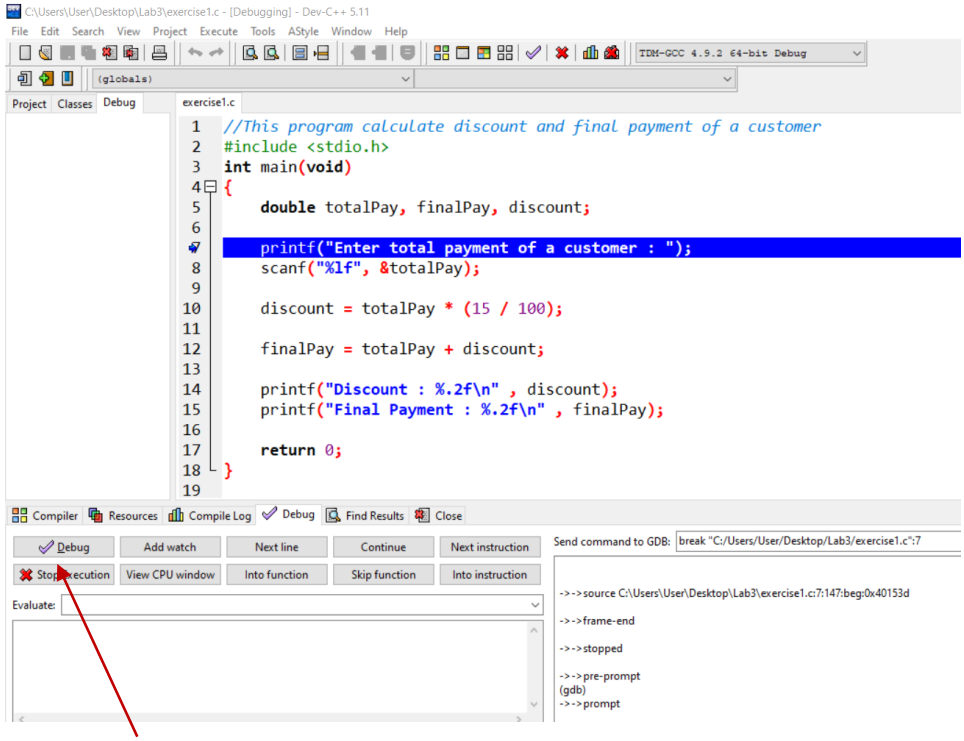
```c
//This program calculate discount and final payment of a customer
#include <stdio.h>
int main(void)
{
    double totalPay, finalPay, discount;

    printf("Enter total payment of a customer : ");
    scanf("%lf", &totalPay);

    discount = totalPay * (15 / 100);

    finalPay = totalPay + discount;

    printf("Discount : %.2f\n" , discount);
    printf("Final Payment : %.2f\n" , finalPay);

    return 0;
}
```

Step 03

Start debugging



Then, debug window will appear as follows.

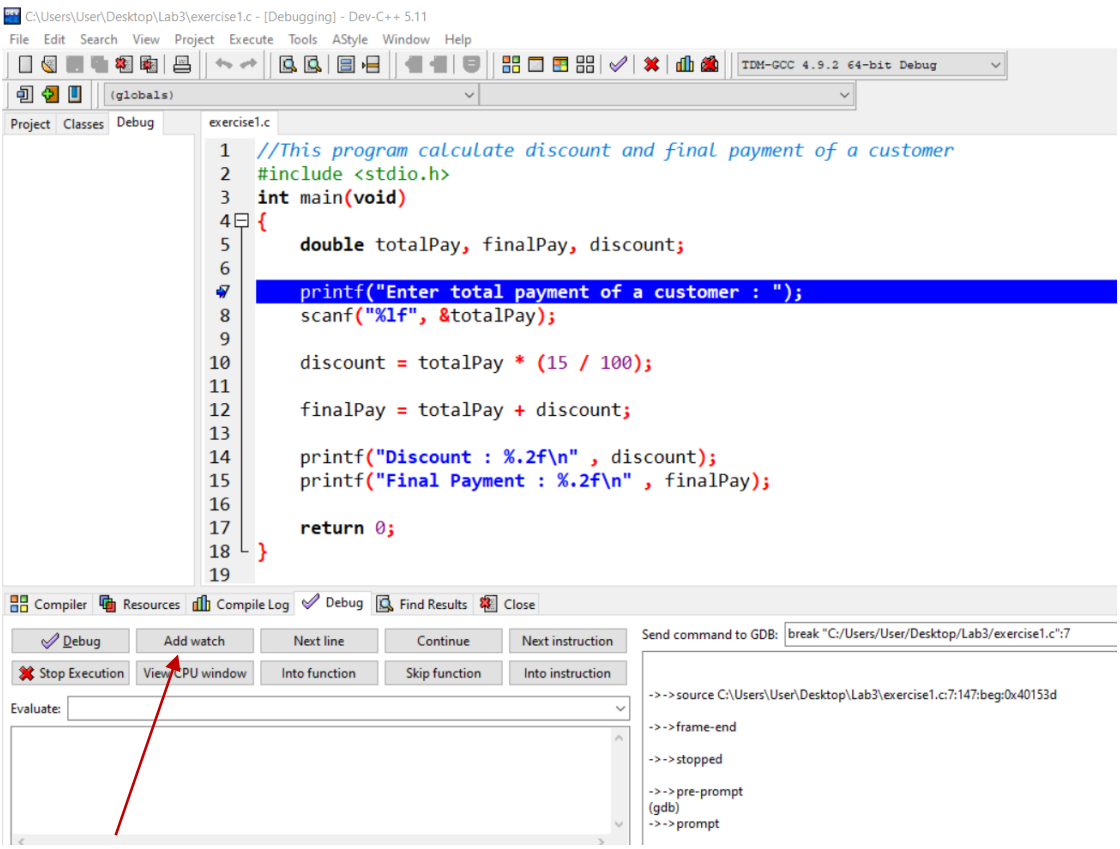


Click on **Debug** button, then your program will be executed up to **line no. 6**

Now `totalPay`, `finalPay` and `discount` variables will be declared and not initialized, but you can check what are the values (garbage values) that are stored in these memory locations.

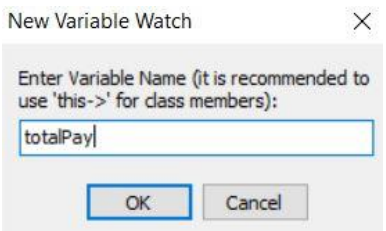
Step 04

Add a watch on a variable



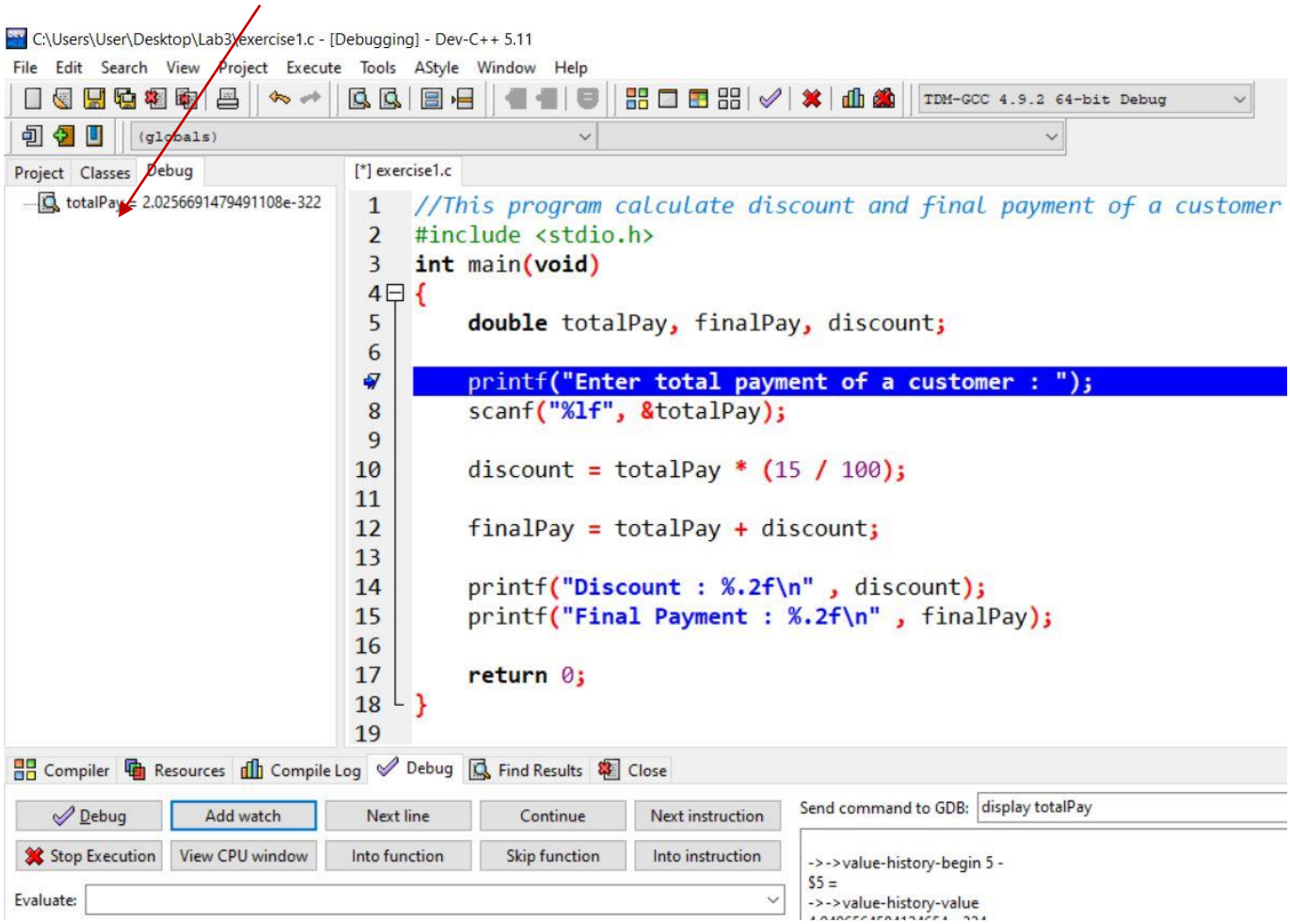
Click on **Add watch** button

Then a pop box will appear,



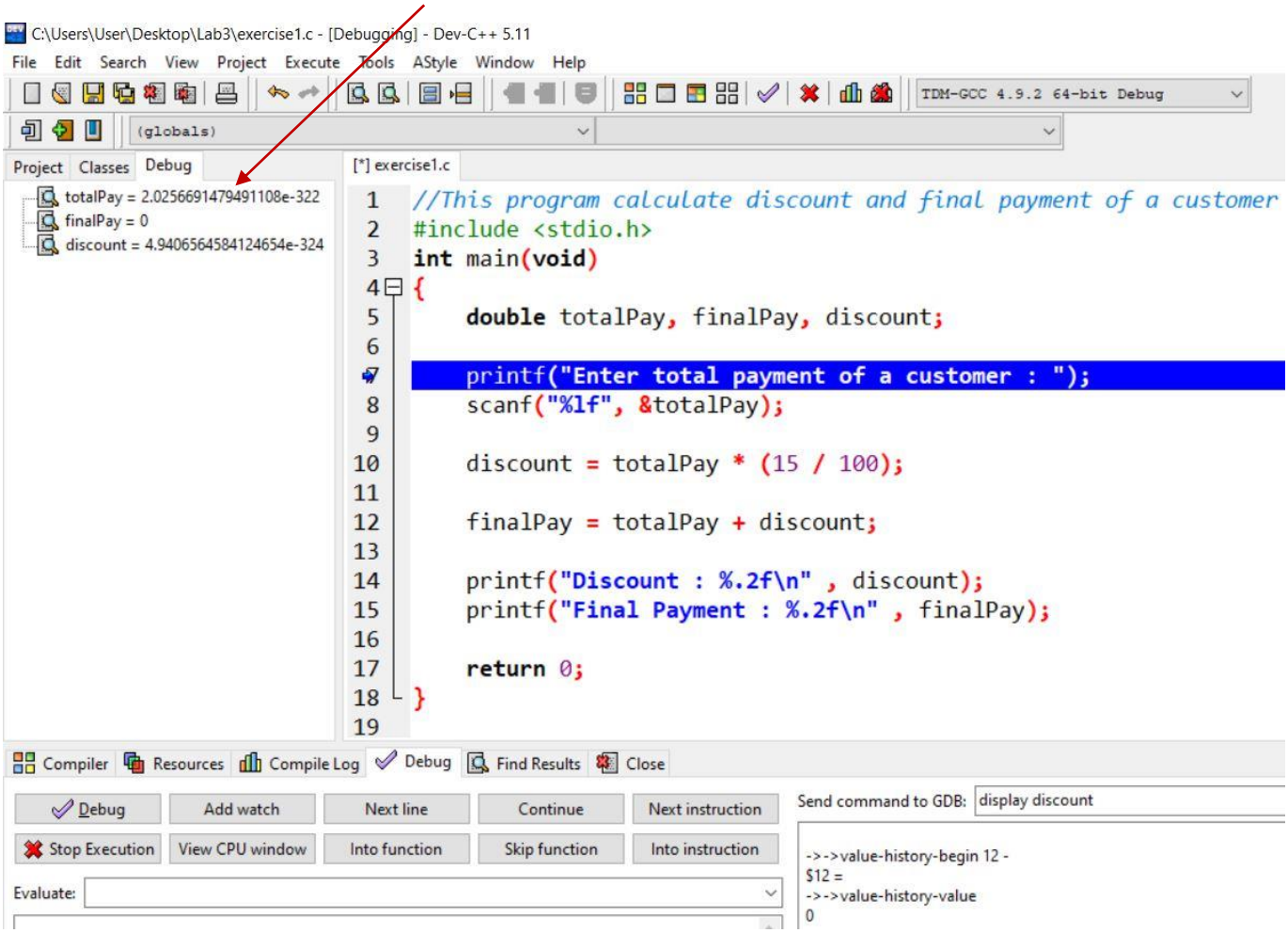
Here, you can give a name of a variable that you are going to add a watch, then click on **OK** button. Now, a watch will be added to `totalPay` variable.

Although we haven't stored any value in this variable, by default **garbage value** is stored in `totalPay` variable.



Step 05

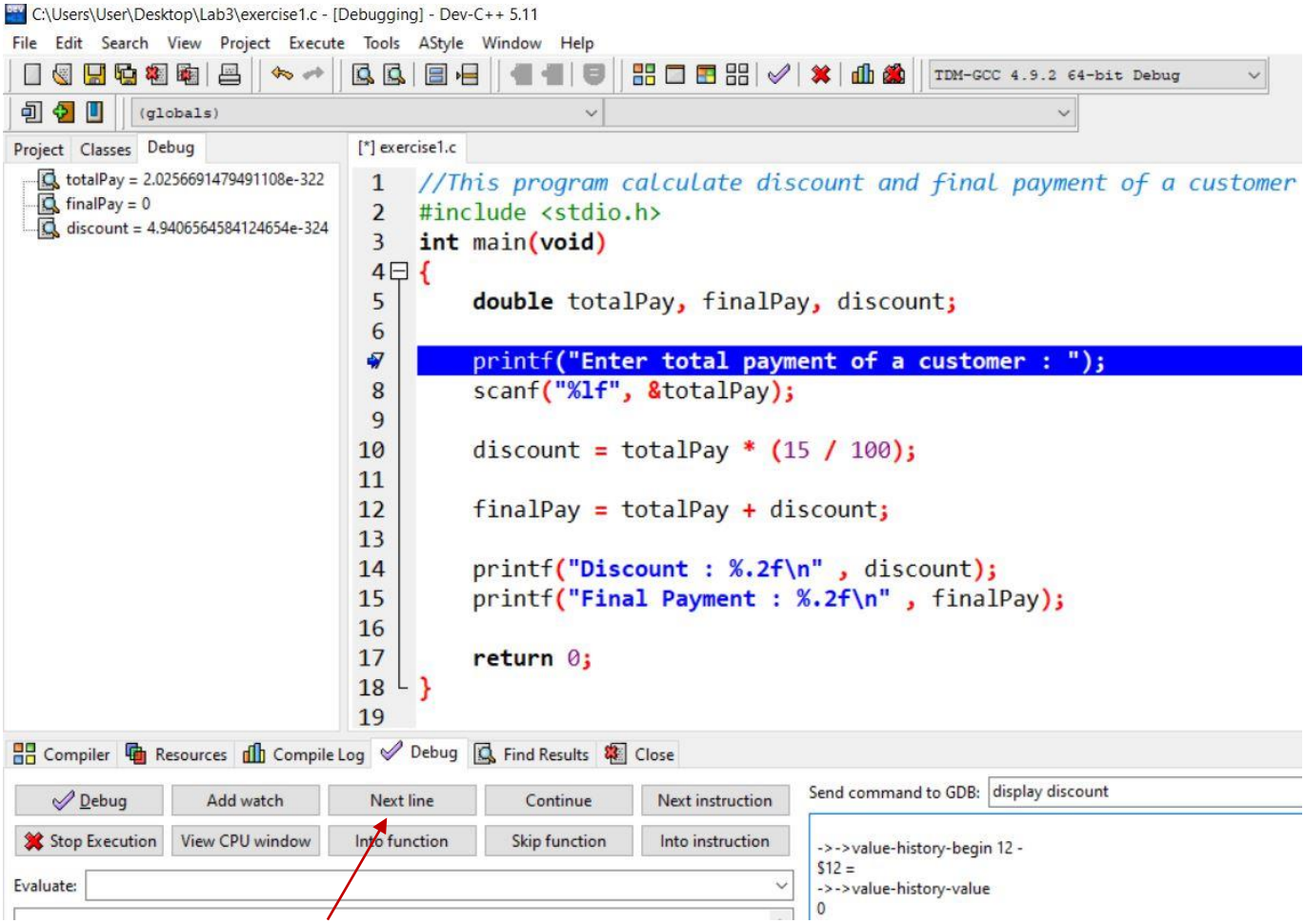
Add watches on other variables and check their values. (`finalPay` , `discount`)



Now, the program is executed up to **line no. 6**.

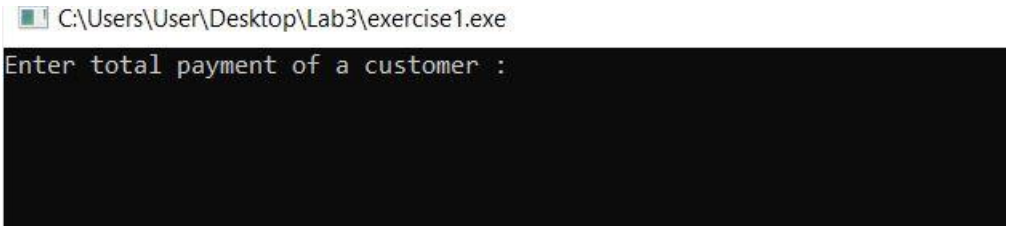
Step 06

To execute next statement in line no. 7, click on **Next line** button.



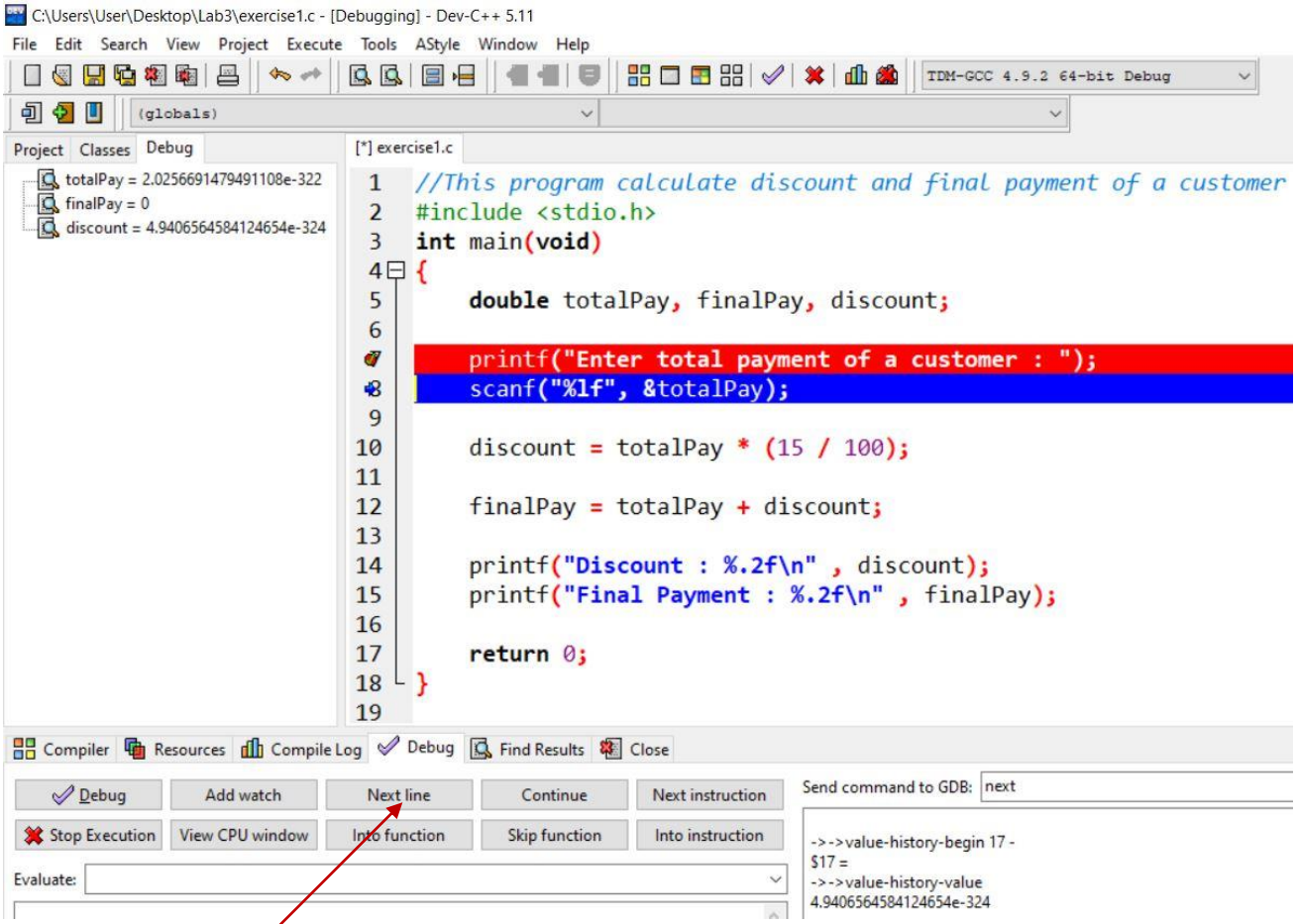
Click on **Next line** button

Then, **line no. 7** will be executed. Now you can see the output window as follows.



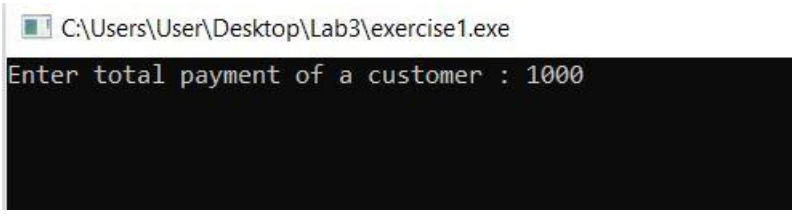
Step 07

To execute next statement line in no. 8, click on **Next line** button again.



Click on **Next line** button

Then, **line no. 8** will be executed. Now you can input total payment of a customer as 1000. Then, press **enter** button in your keyboard.

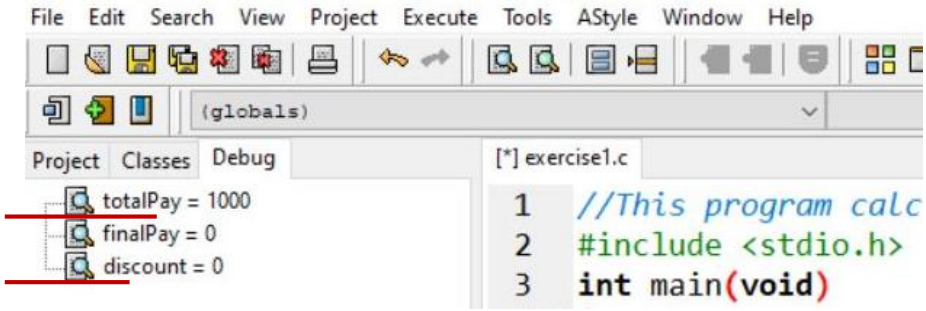


Step 08

To execute next statement in line no. 10, click on **Next line** button again.

Then, **line no. 10** will be executed, and discount will be calculated. The calculated discount value will be stored in `discount` variable.

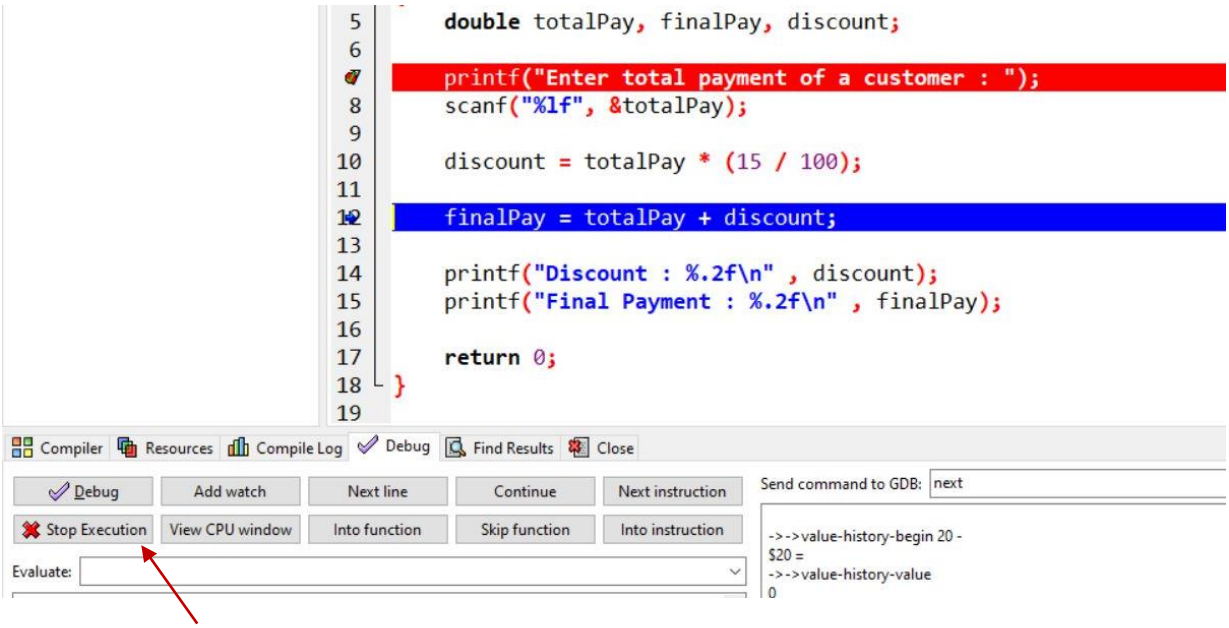
Now, you can see that the variable values of `totalPay` and `discount` are changed to 1000 and 0.



If the user input 1000 from the keyboard for total payment , the calculated discount value can't be zero. It means that there is a logical error in this statement in line no. 10.

Step 09

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



Click on **Stop Execution** button

Step 10

When we observe the statement in line no. 10.

```
discount = totalPay * (15 / 100);
```

Here, we have an integer division .

```
15 / 100
```

The output of this integer by integer division is zero. Then, the value of `totalPay` variable will be multiplied by zero. Because of that, zero will be the output of this expression and zero will be stored in `discount` variable.

Now, you can modify the statement as bellow. (There are several methods to correct this expression. You can use one of those methods.)

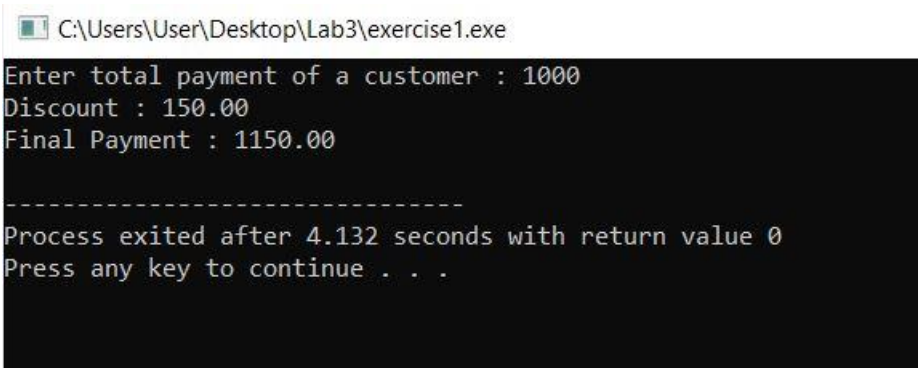
[*] exercise1.c

```
1  //This program calculate discount and final payment of a customer
2  #include <stdio.h>
3  int main(void)
4  {
5      double totalPay, finalPay, discount;
6
7      printf("Enter total payment of a customer : ");
8      scanf("%lf", &totalPay);
9
10     discount = totalPay * (15 / 100.0);
11
12     finalPay = totalPay + discount;
13
14     printf("Discount : %.2f\n" , discount);
15     printf("Final Payment : %.2f\n" , finalPay);
16
17     return 0;
18 }
```

Step 11
Click on the line number of the relevant statement which includes the break point to remove it.

Step 12
Compile and run the program and see whether the program works as expected.

The output of your program will look like the following after fixing the logical error.



If you have entered the total payment as 1000, the discount value should be 150 and final payment value should be 850.

Here, you can see that discount value has displayed correctly but, the final payment value is incorrect.

It means, there may be more logical errors in this C program.
Then you need to debug your program again to identify those logical errors.

4. Now, you need to use debugging option again to find other logical errors in the program.

Step 01

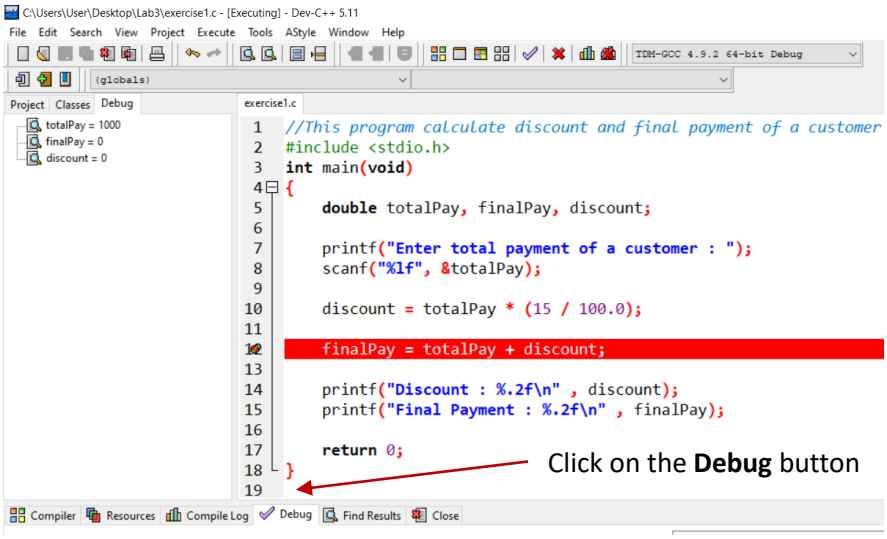
Set a break point in C program

Hint **: When you're going to debug your program again, you can set break point at line no. 12 since in earlier process, we have confirmed that up to line no. 10, there are no logical errors.**

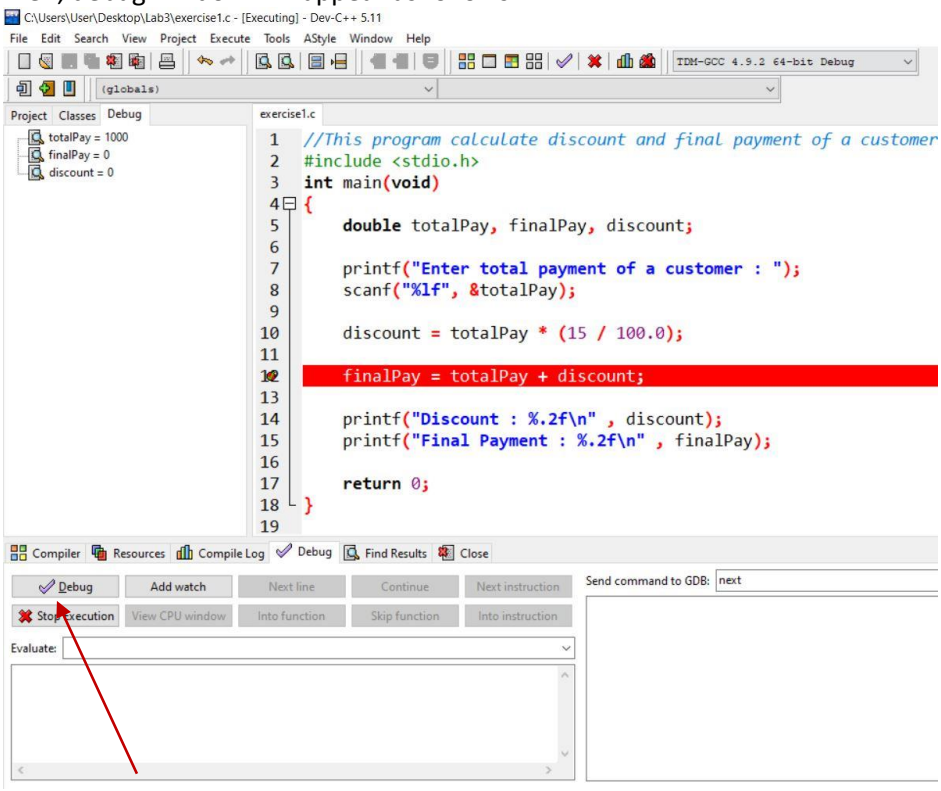
```
1  //This program calculate discount and final payment of a customer
2  #include <stdio.h>
3  int main(void)
4  {
5      double totalPay, finalPay, discount;
6
7      printf("Enter total payment of a customer : ");
8      scanf("%lf", &totalPay);
9
10     discount = totalPay * (15 / 100.0);
11
12     finalPay = totalPay + discount;
13
14     printf("Discount : %.2f\n" , discount);
15     printf("Final Payment : %.2f\n" , finalPay);
16
17     return 0;
18 }
```

Step 02

Start debugging



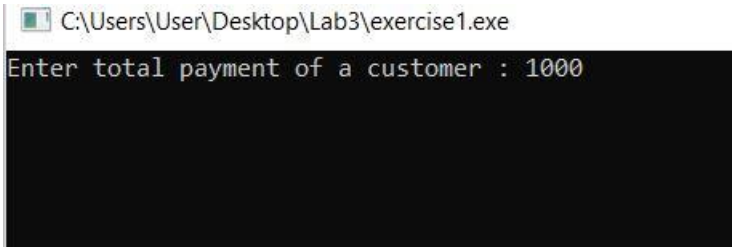
Then, debug window will appear as follows.



Click on **Debug** button, then your program will be executed up to **line no. 10**

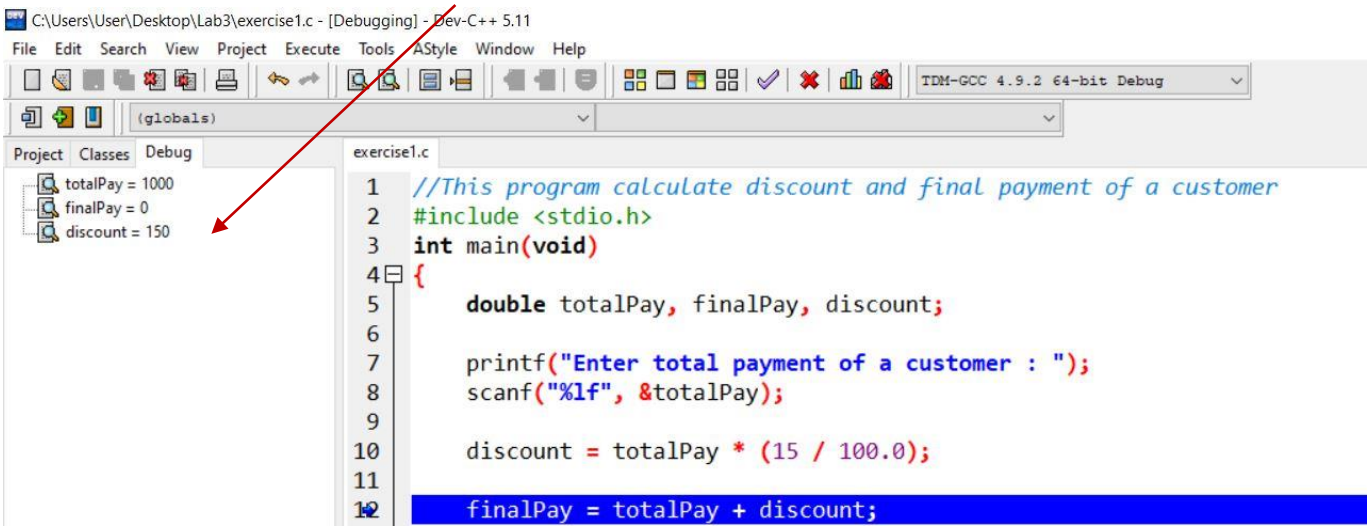
Step 03

In output window, you can input total payment of a customer as 1000. Then, the discount will be calculated since the program is executed up to **line no. 10**



Step 04

The value of `discount` variable is changed to 150.

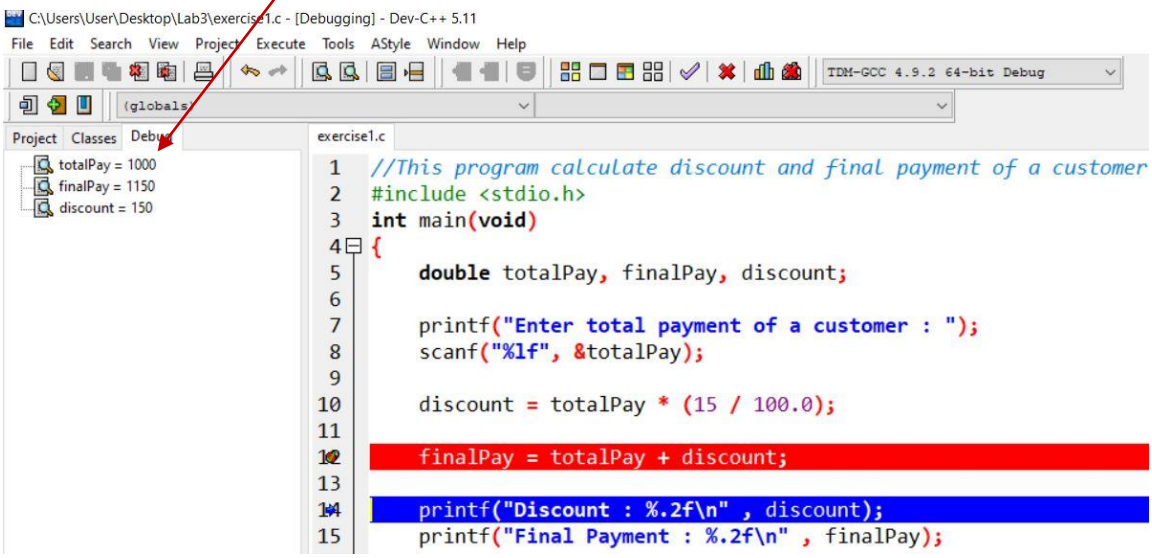


Now, the program is executed up to **line no. 10**.

Step 05

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

Then, **line no. 12** will be executed, and the final payment will be calculated. Now you can see the value of `finalPay` variable as 1150. But, it's incorrect. It means that in this line, there can be a logical error.



Step 06

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

Step 07

When we observe the statement in line no. 12.

```
finalPay = totalPay + discount;
```

We can see that the discount value is added to the total payment to calculate the final payment. But the discount value should be subtracted from the total payment to calculate the final payment.

Now, you can modify the statement as bellow.

```c
//This program calculate discount and final payment of a customer
#include <stdio.h>
int main(void)
{
    double totalPay, finalPay, discount;

    printf("Enter total payment of a customer : ");
    scanf("%lf", &totalPay);

    discount = totalPay * (15 / 100.0);

    finalPay = totalPay - discount;

    printf("Discount : %.2f\n" , discount);
    printf("Final Payment : %.2f\n" , finalPay);

    return 0;
}
```

Step 08

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

Step 09

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

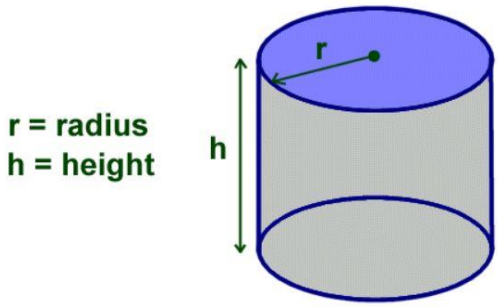
If you can't get the expected output, there may be more logical errors. Then you need to continue debugging your program to identify those logical errors.

Exercise 2

Use following sample program and identify **the logical errors** in it using debugging option.

This program calculates the surface area of cylinder (A).

$$A = 2\pi r h + 2\pi r^2$$




```
//This program calculate discount and final payment of a customer
#include <stdio.h>
int main(void)
{
      float  r, h, areaRec, areaCircle, area;

      printf("Enter radius of the Cylinder : ");
      scanf("%f", &r);

      printf("Enter height of the Cylinder : ");
      scanf("%f", &h);

      areaRec = 2 * 22 / 7 * r * h;
      areaCircle = 22 / 7 * r * r;

      area = areaRec + areaCircle;

      printf("Surface area of cylinder : %.1f\n", area);

      return 0;
}
```

Sample input/output:

Enter radius of the Cylinder : 7
Enter height of the Cylinder : 10
Surface area of cylinder : 748.0