Debug

Last updated 6/19/23

Debug

- Easy Debug Tactics
  - Print out intermediate information

```
printf(“I reached this point”);
printf(“foo = %i\n”, foo);
```

  - Break problems into pieces

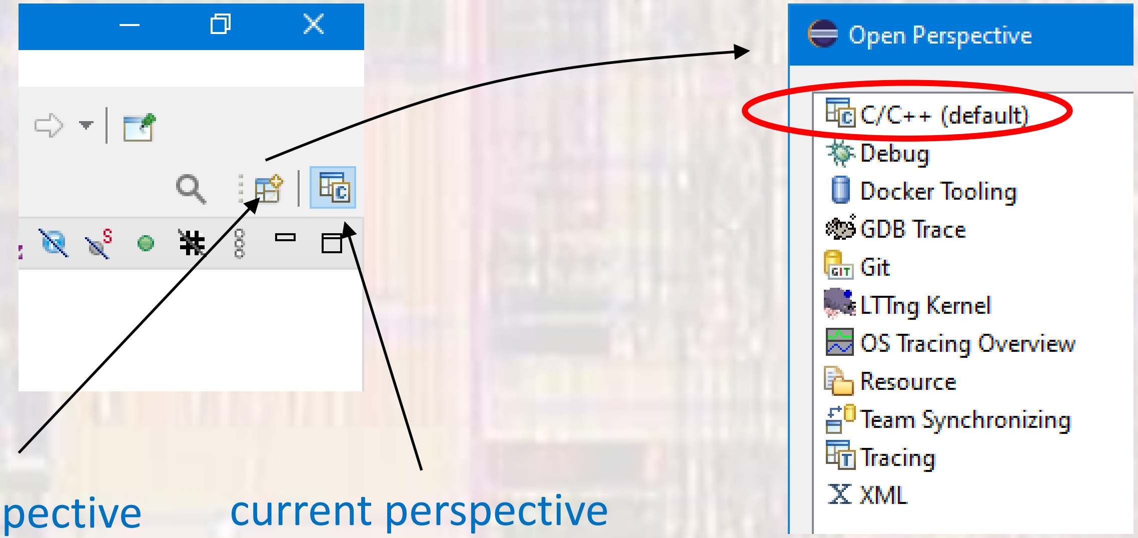
```
foo = a | b << c * d++ - 3 /b % 6;
        →
foo = d++;
printf(“foo = %i\n”, foo);
foo = c * d++ ;
printf(“foo = %i\n”, foo);
…
```

Debug

- Debugger
  - Most C tool chains include a debugger
  - The debugger allows
    - Stopping execution
    - Stepping line – by – line
    - Tracking variable values
    - Follow execution into and out of functions

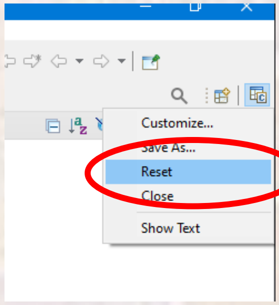
Debug

- Debugger - Eclipse Perspectives
  - Eclipse has a series of pre-defined window configurations
  - Each configuration is optimized for a specific purpose



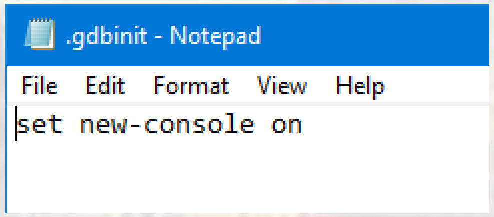
select perspective   current perspective

- If your windows get messed up
  - rt-click on the current perspective → Reset



Debug

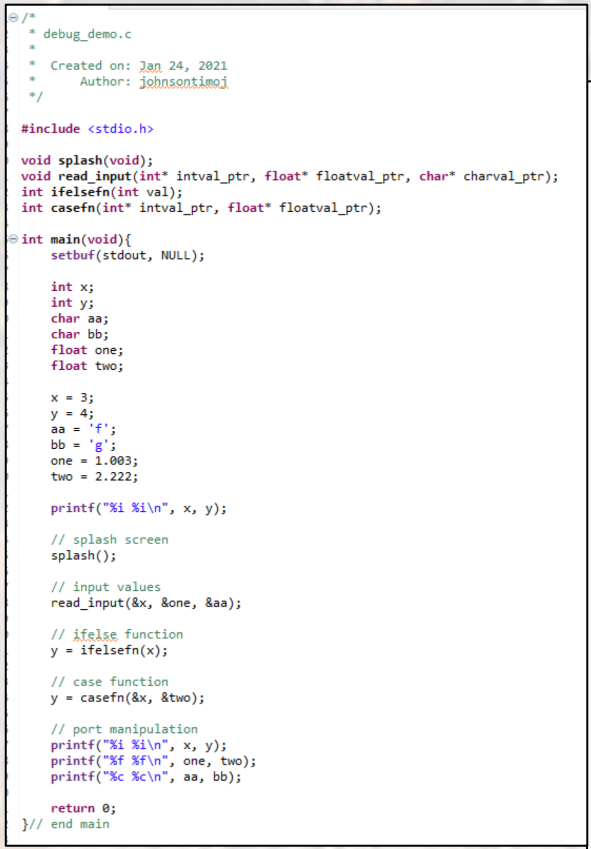
- Eclipse Debugger – Work-around
  - The debugger does not work in the UI console window
    - To work-around this issue we will use an external terminal window
  - rt-click the project name New → File
  - provide the file name .gdbinit
  - In the opened file type set new-console on
    - save
    - Note – the file will not show up in the Project Explorer list



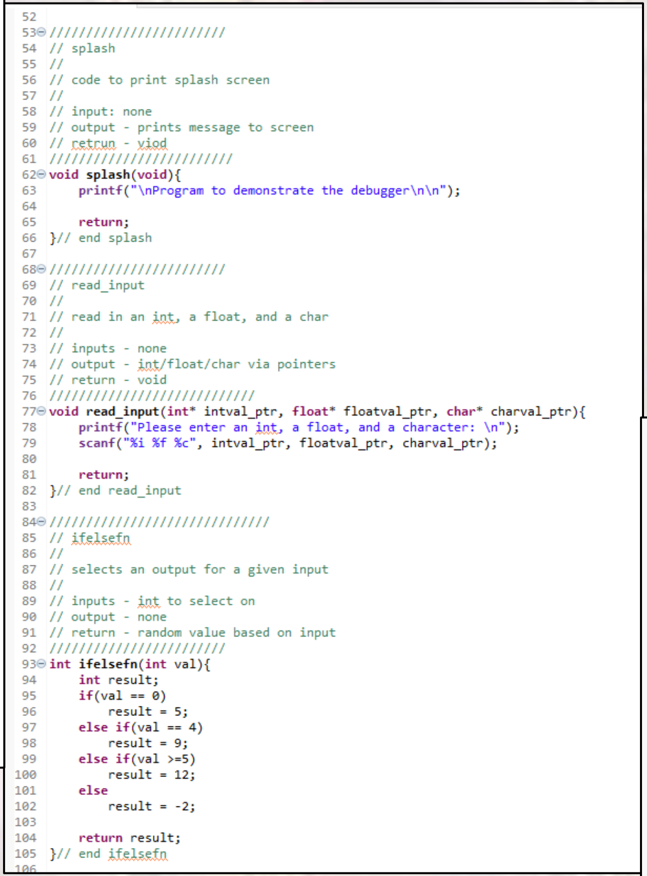
```
set new-console on
```

Debug

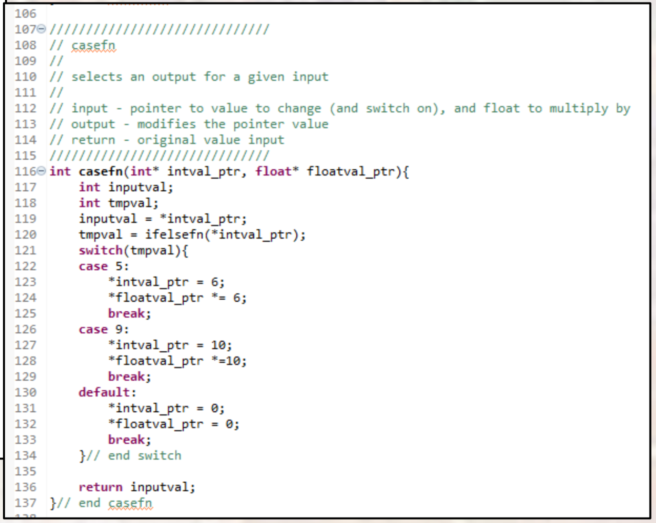
- Debugger Tool - Example program



```c
/*
 * debug_demo.c
 *
 *  Created on: Jan 24, 2021
 *      Author: johnsontimoj
 */

#include <stdio.h>

void splash(void);
void read_input(int* intval_ptr, float* floatval_ptr, char* charval_ptr);
int ifelsefn(int val);
int casefn(int* intval_ptr, float* floatval_ptr);

int main(void){
    setbuf(stdout, NULL);

    int x;
    int y;
    char aa;
    char bb;
    float one;
    float two;

    x = 3;
    y = 4;
    aa = 'f';
    bb = 'g';
    one = 1.003;
    two = 2.222;

    printf("%i %i\n", x, y);

    // splash screen
    splash();

    // input values
    read_input(&x, &one, &aa);

    // ifelse function
    y = ifelsefn(x);

    // case function
    y = casefn(&x, &two);

    // port manipulation
    printf("%i %i\n", x, y);
    printf("%f %f\n", one, two);
    printf("%c %c\n", aa, bb);

    return 0;
}// end main
```



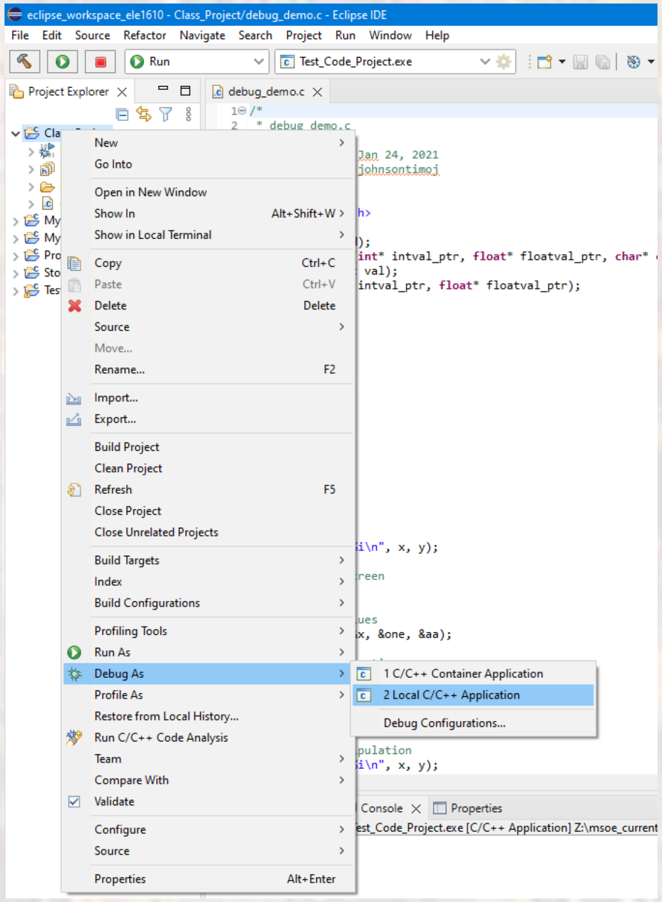
```c

////////////////////////
// splash
//
// code to print splash screen
//
// input: none
// output - prints message to screen
// retrun - viod
/////////////////////////
void splash(void){
    printf("\nProgram to demonstrate the debugger\n\n");

    return;
}// end splash

////////////////////////
// read_input
//
// read in an int, a float, and a char
//
// inputs - none
// output - int/float/char via pointers
// return - void
////////////////////////////
void read_input(int* intval_ptr, float* floatval_ptr, char* charval_ptr){
    printf("Please enter an int, a float, and a character: \n");
    scanf("%i %f %c", intval_ptr, floatval_ptr, charval_ptr);

    return;
}// end read_input

/////////////////////////////
// ifelsefn
//
// selects an output for a given input
//
// inputs - int to select on
// output - none
// return - random value based on input
////////////////////////
int ifelsefn(int val){
    int result;
    if(val == 0)
        result = 5;
    else if(val == 4)
        result = 9;
    else if(val >=5)
        result = 12;
    else
        result = -2;

    return result;
}// end ifelsefn
```



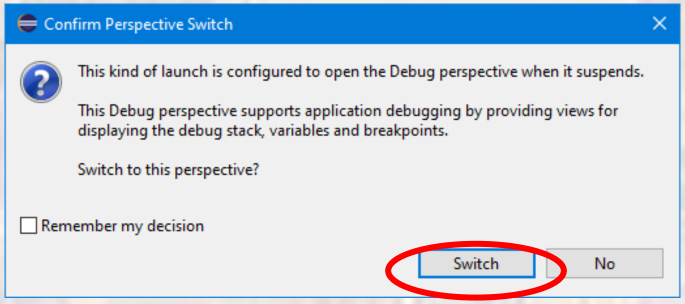
```c

//////////////////////////////
// casefn
//
// selects an output for a given input
//
// input - pointer to value to change (and switch on), and float to multiply by
// output - modifies the pointer value
// return - original value input
//////////////////////////////
int casefn(int* intval_ptr, float* floatval_ptr){
    int inputval;
    int tmpval;
    inputval = *intval_ptr;
    tmpval = ifelsefn(*intval_ptr);
    switch(tmpval){
    case 5:
        *intval_ptr = 6;
        *floatval_ptr *= 6;
        break;
    case 9:
        *intval_ptr = 10;
        *floatval_ptr *=10;
        break;
    default:
        *intval_ptr = 0;
        *floatval_ptr = 0;
        break;
    }// end switch

    return inputval;
}// end casefn
```

Debug

- Eclipse Debugger
  - rt-click on your project → Debug As → Local C/C++ Application



  - Select Switch to change the perspective



This kind of launch is configured to open the Debug perspective when it suspends.

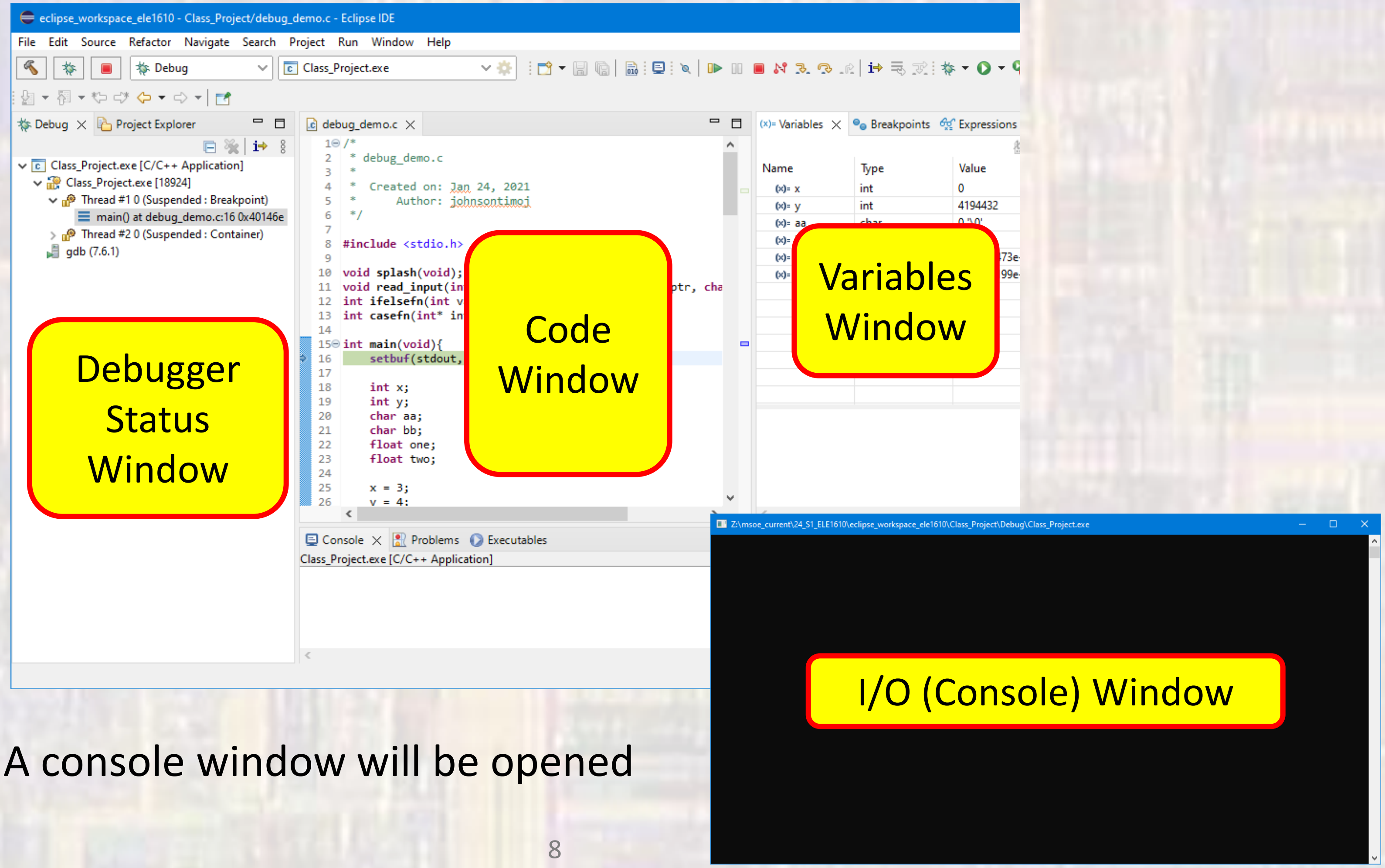
This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints.

Switch to this perspective?

☐ Remember my decision

Debug

- Eclipse Debugger
  - The Eclipse perspective will be changed


eclipse_workspace_ele1610 - Class_Project/debug_demo.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Class_Project.exe
Debug
Project Explorer
debug_demo.c
(x)= Variables
Breakpoints
Expressions
Console
Problems
Executables
Z:\msoe_current\24_S1_ELE1610\eclipse_workspace_ele1610\Class_Project\Debug\Class_Project.exe
Debugger Status Window
Code Window
Variables Window
I/O (Console) Window

```
Class_Project.exe [C/C++ Application]
  Class_Project.exe [18924]
    Thread #1 0 (Suspended : Breakpoint)
      main() at debug_demo.c:16 0x40146e
    Thread #2 0 (Suspended : Container)
  gdb (7.6.1)
```

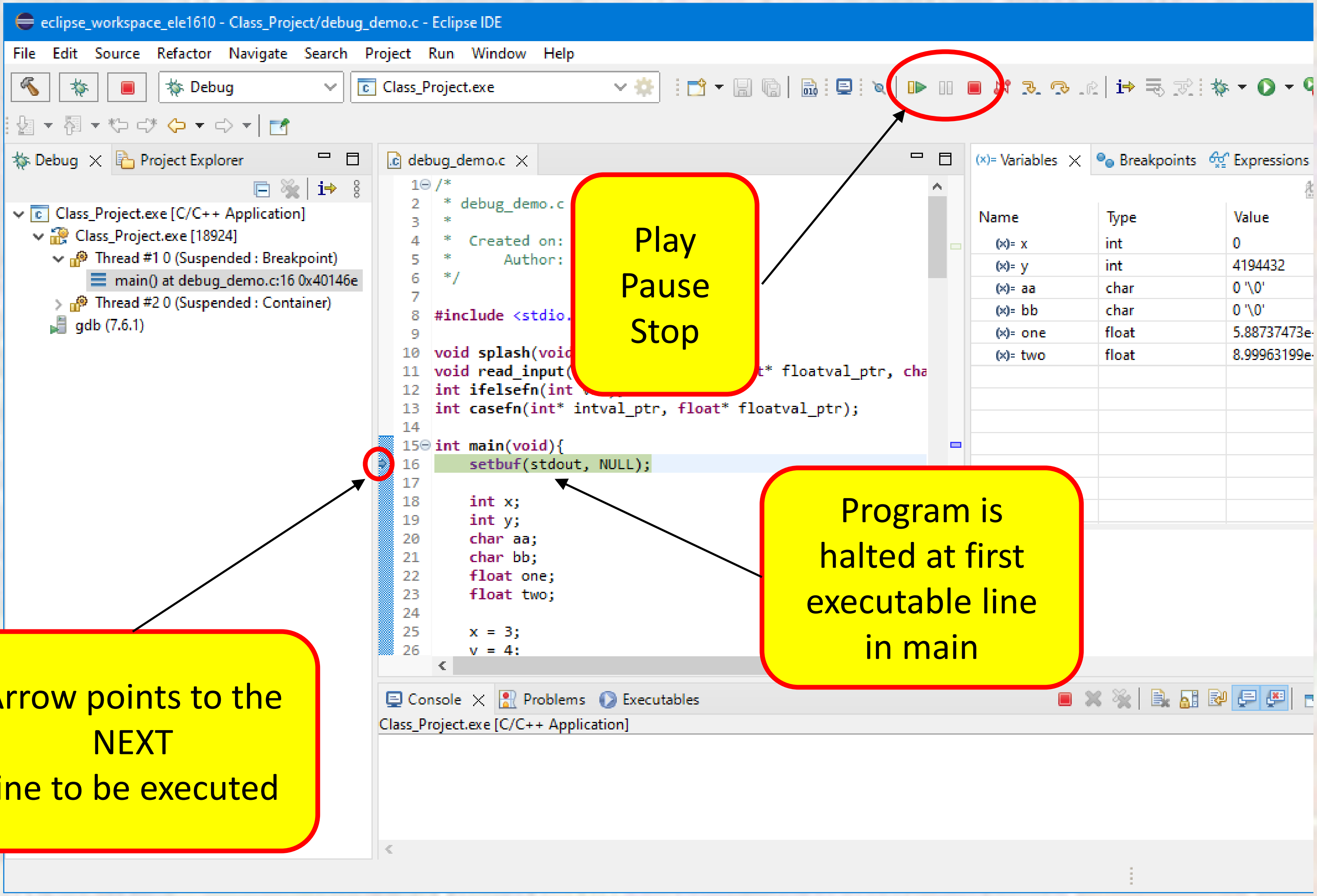
```
/*
 * debug_demo.c
 *
 *  Created on: Jan 24, 2021
 *      Author: johnsontimoj
 */

#include <stdio.h>

void splash(void);
void read_input(in                                  ptr, cha
int ifelsefn(int v
int casefn(int* in

int main(void){
    setbuf(stdout,

    int x;
    int y;
    char aa;
    char bb;
    float one;
    float two;

    x = 3;
    y = 4;
```

Name	Type	Value
x	int	0
y	int	4194432
aa	char	0 '\0'

```
Class_Project.exe [C/C++ Application]
```

  - A console window will be opened

Debug


eclipse_workspace_ele1610 - Class_Project/debug_demo.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Class_Project.exe
Debug
Project Explorer
debug_demo.c
Variables
Breakpoints
Expressions
Console
Problems
Executables

```
Class_Project.exe [C/C++ Application]
  Class_Project.exe [18924]
    Thread #1 0 (Suspended : Breakpoint)
      main() at debug_demo.c:16 0x40146e
    Thread #2 0 (Suspended : Container)
  gdb (7.6.1)
```

```
/*
 * debug_demo.c
 *
 *  Created on:
 *      Author:
 */

#include <stdio.

void splash(voi
void read_input(                    t* floatval_ptr, cha
int ifelsefn(int
int casefn(int* intval_ptr, float* floatval_ptr);

int main(void){
    setbuf(stdout, NULL);

    int x;
    int y;
    char aa;
    char bb;
    float one;
    float two;

    x = 3;
    y = 4;
```

Name	Type	Value
(x)= x	int	0
(x)= y	int	4194432
(x)= aa	char	0 '\0'
(x)= bb	char	0 '\0'
(x)= one	float	5.88737473e-
(x)= two	float	8.99963199e-

```
Class_Project.exe [C/C++ Application]
```

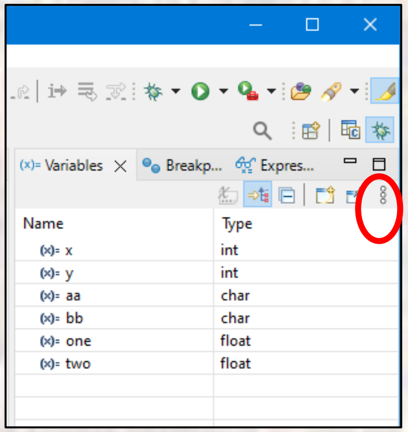
Play
Pause
Stop

Program is halted at first executable line in main

Arrow points to the NEXT line to be executed

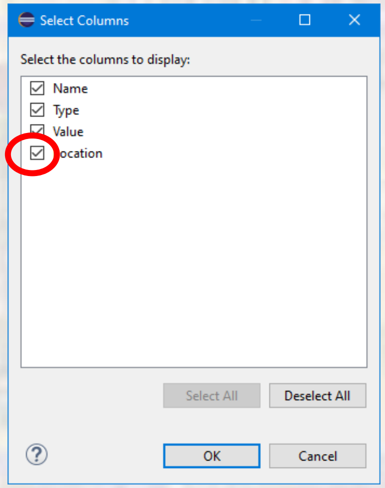
Debug

- Eclipse Debugger – default data change
  - The debugger defaults to not-showing the memory location of variables
    - To modify this to show the memory location of variables
    - Click the 3-vertical-dots in the variable window



Name	Type
x	int
y	int
aa	char
bb	char
one	float
two	float

    - Select Layout → Select Columns
    - Check the Location box

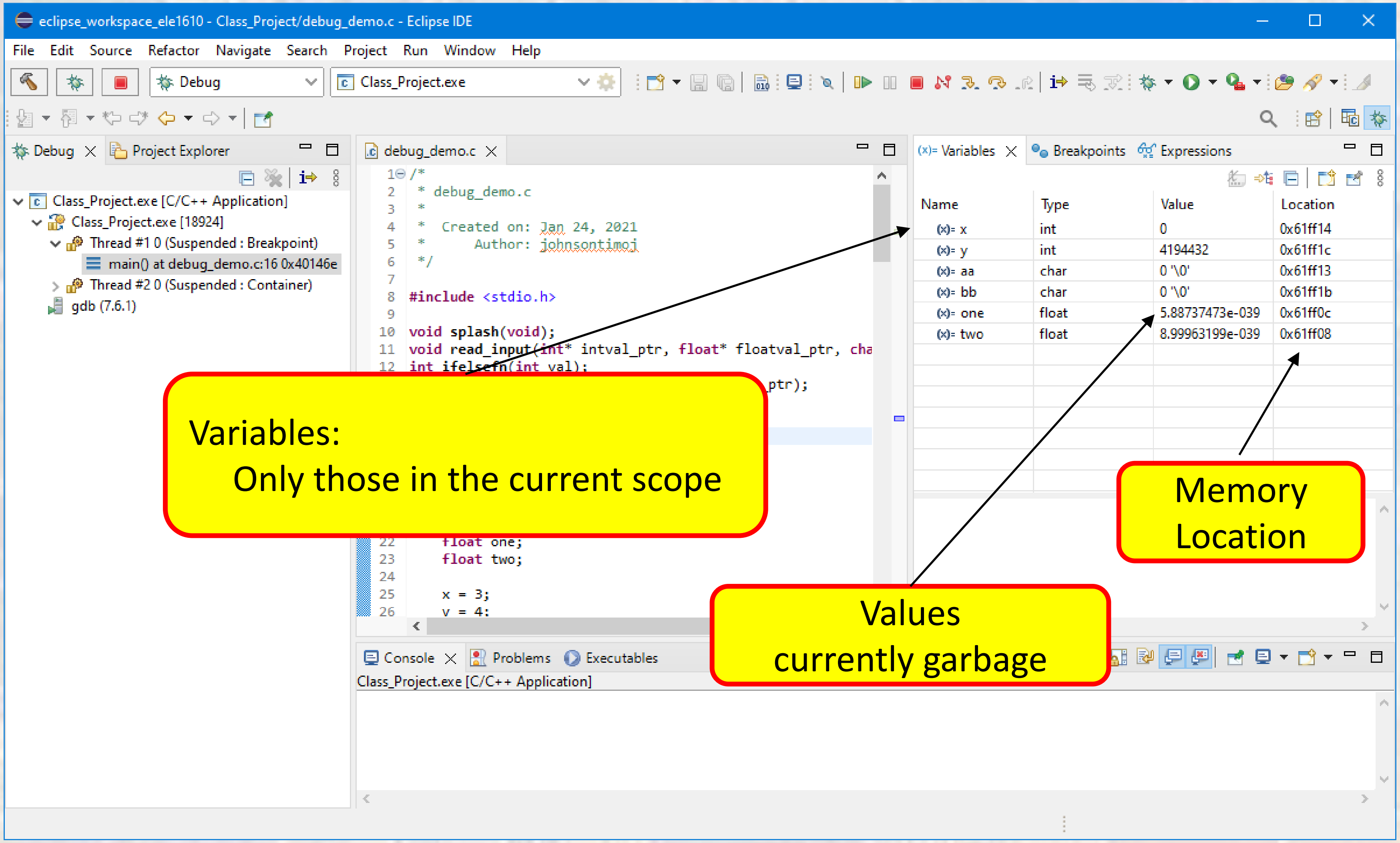


Select Columns

Select the columns to display:

- [✓] Name
- [✓] Type
- [✓] Value
- [✓] Location

Debug



eclipse_workspace_ele1610 - Class_Project/debug_demo.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Class_Project.exe
Debug
Project Explorer
debug_demo.c
Variables
Breakpoints
Expressions
Console
Problems
Executables

```
Class_Project.exe [C/C++ Application]
  Class_Project.exe [18924]
    Thread #1 0 (Suspended : Breakpoint)
      main() at debug_demo.c:16 0x40146e
    Thread #2 0 (Suspended : Container)
  gdb (7.6.1)
```

```
/*
 * debug_demo.c
 *
 *  Created on: Jan 24, 2021
 *      Author: johnsontimoj
 */

#include <stdio.h>

void splash(void);
void read_input(int* intval_ptr, float* floatval_ptr, cha
int ifelsefn(int val);
                                                  ptr);

    float one;
    float two;

    x = 3;
    y = 4;
```

Name	Type	Value	Location
(x)= x	int	0	0x61ff14
(x)= y	int	4194432	0x61ff1c
(x)= aa	char	0 '\0'	0x61ff13
(x)= bb	char	0 '\0'	0x61ff1b
(x)= one	float	5.88737473e-039	0x61ff0c
(x)= two	float	8.99963199e-039	0x61ff08

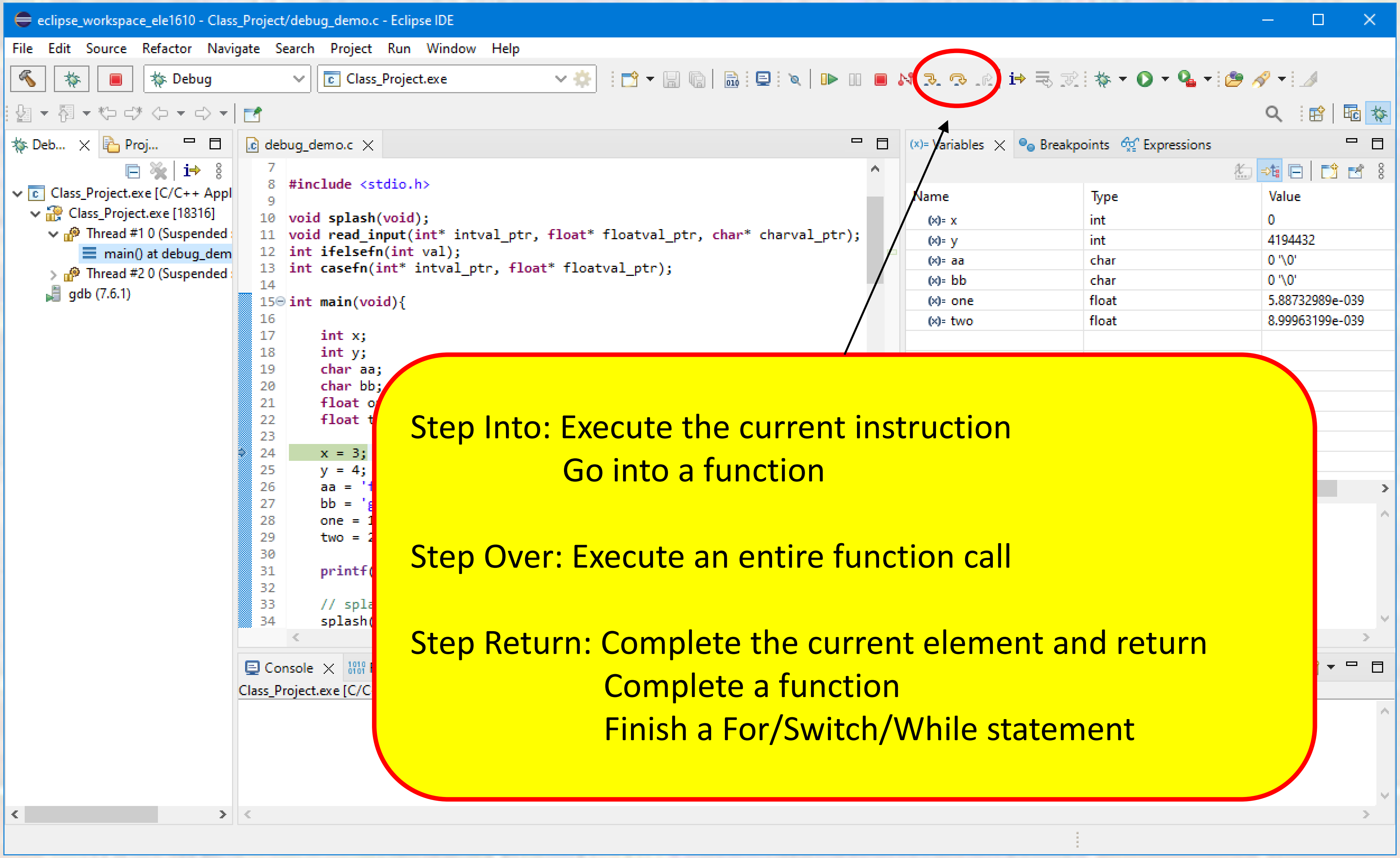
```
Class_Project.exe [C/C++ Application]
```

Variables:
Only those in the current scope

Values
currently garbage

Memory
Location

Debug


eclipse_workspace_ele1610 - Class_Project/debug_demo.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Class_Project.exe
Deb... Proj...
debug_demo.c
Variables Breakpoints Expressions
Console

```
Class_Project.exe [C/C++ Appl
  Class_Project.exe [18316]
    Thread #1 0 (Suspended
      main() at debug_dem
    Thread #2 0 (Suspended
  gdb (7.6.1)
```

```
#include <stdio.h>

void splash(void);
void read_input(int* intval_ptr, float* floatval_ptr, char* charval_ptr);
int ifelsefn(int val);
int casefn(int* intval_ptr, float* floatval_ptr);

int main(void){

    int x;
    int y;
    char aa;
    char bb;
    float o
    float t

    x = 3;
    y = 4;
    aa = 'f
    bb = 'g
    one = 1
    two = 2

    printf(

    // spla
    splash(
```

Name	Type	Value
x	int	0
y	int	4194432
aa	char	0 '\0'
bb	char	0 '\0'
one	float	5.88732989e-039
two	float	8.99963199e-039

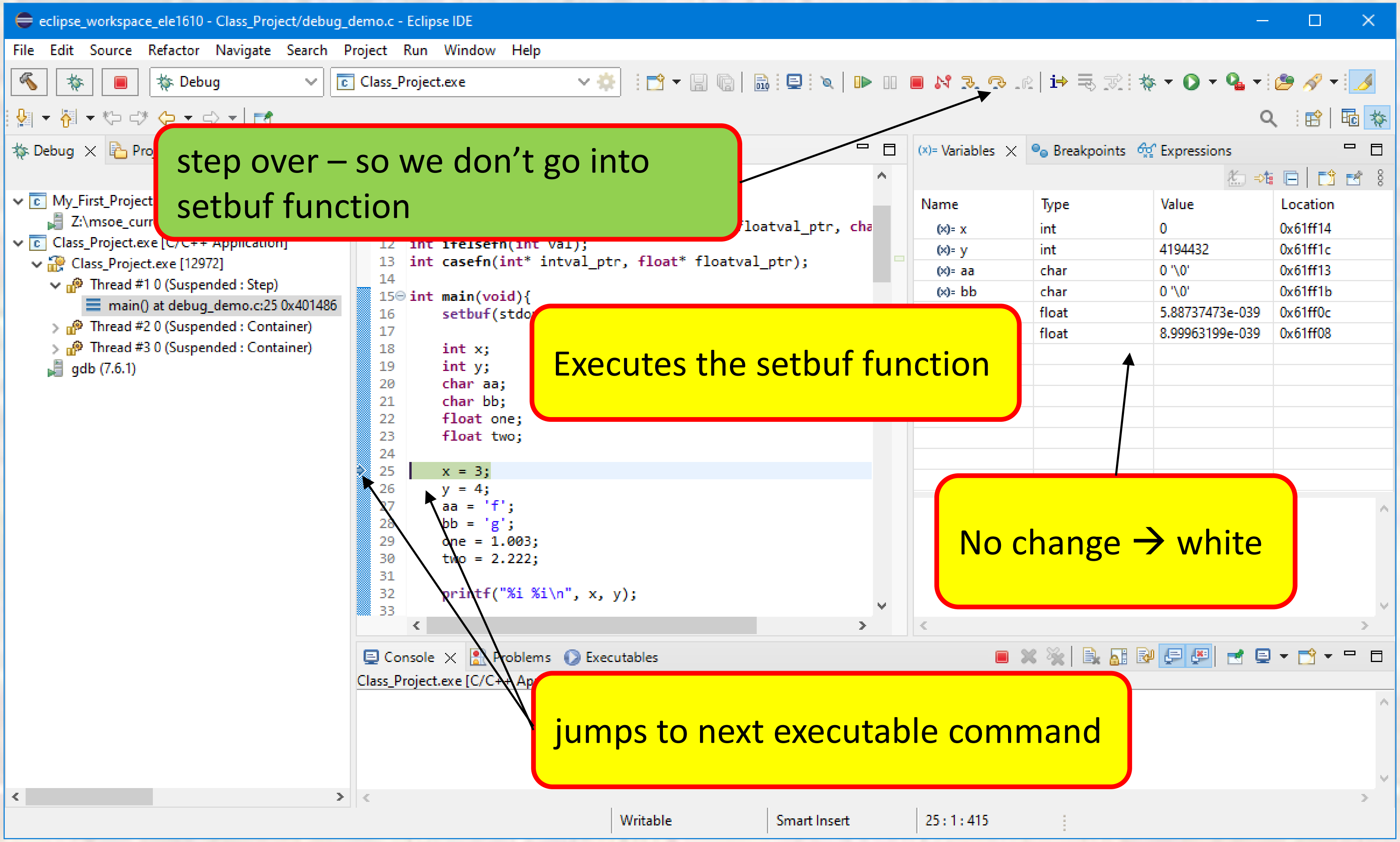
```
Class_Project.exe [C/C
```

Step Into: Execute the current instruction
Go into a function

Step Over: Execute an entire function call

Step Return: Complete the current element and return
Complete a function
Finish a For/Switch/While statement

Debug



eclipse_workspace_ele1610 - Class_Project/debug_demo.c - Eclipse IDE

step over – so we don't go into setbuf function

Executes the setbuf function

No change → white

jumps to next executable command

```
My_First_Project
  Z:\msoe_curr
Class_Project.exe [C/C++ Application]
  Class_Project.exe [12972]
    Thread #1 0 (Suspended : Step)
      main() at debug_demo.c:25 0x401486
    Thread #2 0 (Suspended : Container)
    Thread #3 0 (Suspended : Container)
  gdb (7.6.1)
```

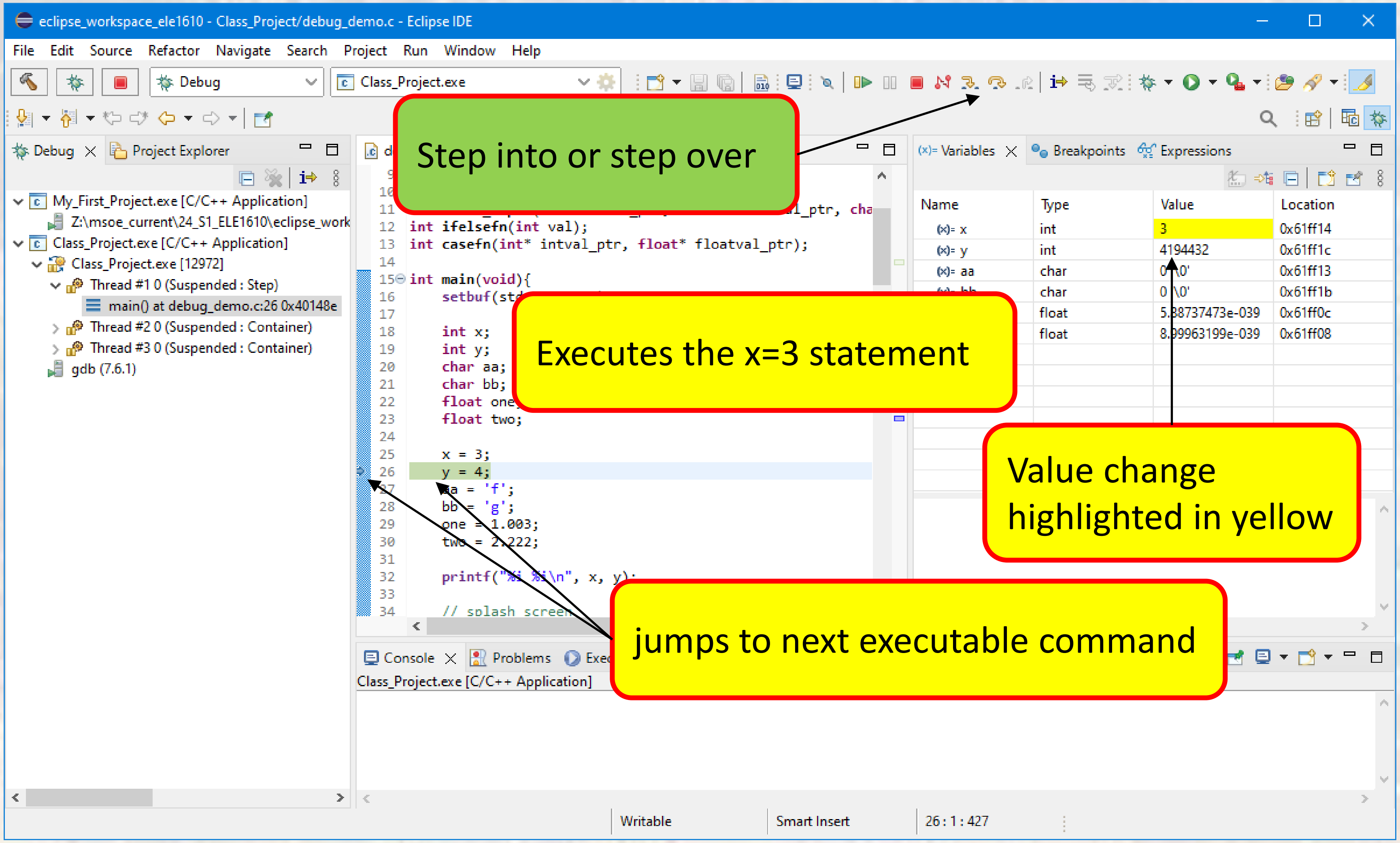
```
                                        floatval_ptr, cha
int ifelsefn(int val);
int casefn(int* intval_ptr, float* floatval_ptr);

int main(void){
    setbuf(stdo

    int x;
    int y;
    char aa;
    char bb;
    float one;
    float two;

    x = 3;
    y = 4;
    aa = 'f';
    bb = 'g';
    one = 1.003;
    two = 2.222;

    printf("%i %i\n", x, y);
```

Name	Type	Value	Location
x	int	0	0x61ff14
y	int	4194432	0x61ff1c
aa	char	0 '\0'	0x61ff13
bb	char	0 '\0'	0x61ff1b
	float	5.88737473e-039	0x61ff0c
	float	8.99963199e-039	0x61ff08

Class_Project.exe [C/C++ Ap

Writable | Smart Insert | 25 : 1 : 415

Debug


eclipse_workspace_ele1610 - Class_Project/debug_demo.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Class_Project.exe
Debug
Project Explorer
Variables
Breakpoints
Expressions
Console
Problems
Writable
Smart Insert
26 : 1 : 427

```
My_First_Project.exe [C/C++ Application]
    Z:\msoe_current\24_S1_ELE1610\eclipse_work
Class_Project.exe [C/C++ Application]
    Class_Project.exe [12972]
        Thread #1 0 (Suspended : Step)
            main() at debug_demo.c:26 0x40148e
        Thread #2 0 (Suspended : Container)
        Thread #3 0 (Suspended : Container)
    gdb (7.6.1)
```

```
                                          _ptr, cha
int ifelsefn(int val);
int casefn(int* intval_ptr, float* floatval_ptr);

int main(void){
    setbuf(std

    int x;
    int y;
    char aa;
    char bb;
    float one
    float two;

    x = 3;
    y = 4;
    aa = 'f';
    bb = 'g';
    one = 1.003;
    two = 2.222;

    printf("%i %i\n", x, y);

    // splash screen
```

Name	Type	Value	Location
x	int	3	0x61ff14
y	int	4194432	0x61ff1c
aa	char	0 '\0'	0x61ff13
bb	char	0 '\0'	0x61ff1b
	float	5.88737473e-039	0x61ff0c
	float	8.99963199e-039	0x61ff08

Class_Project.exe [C/C++ Application]

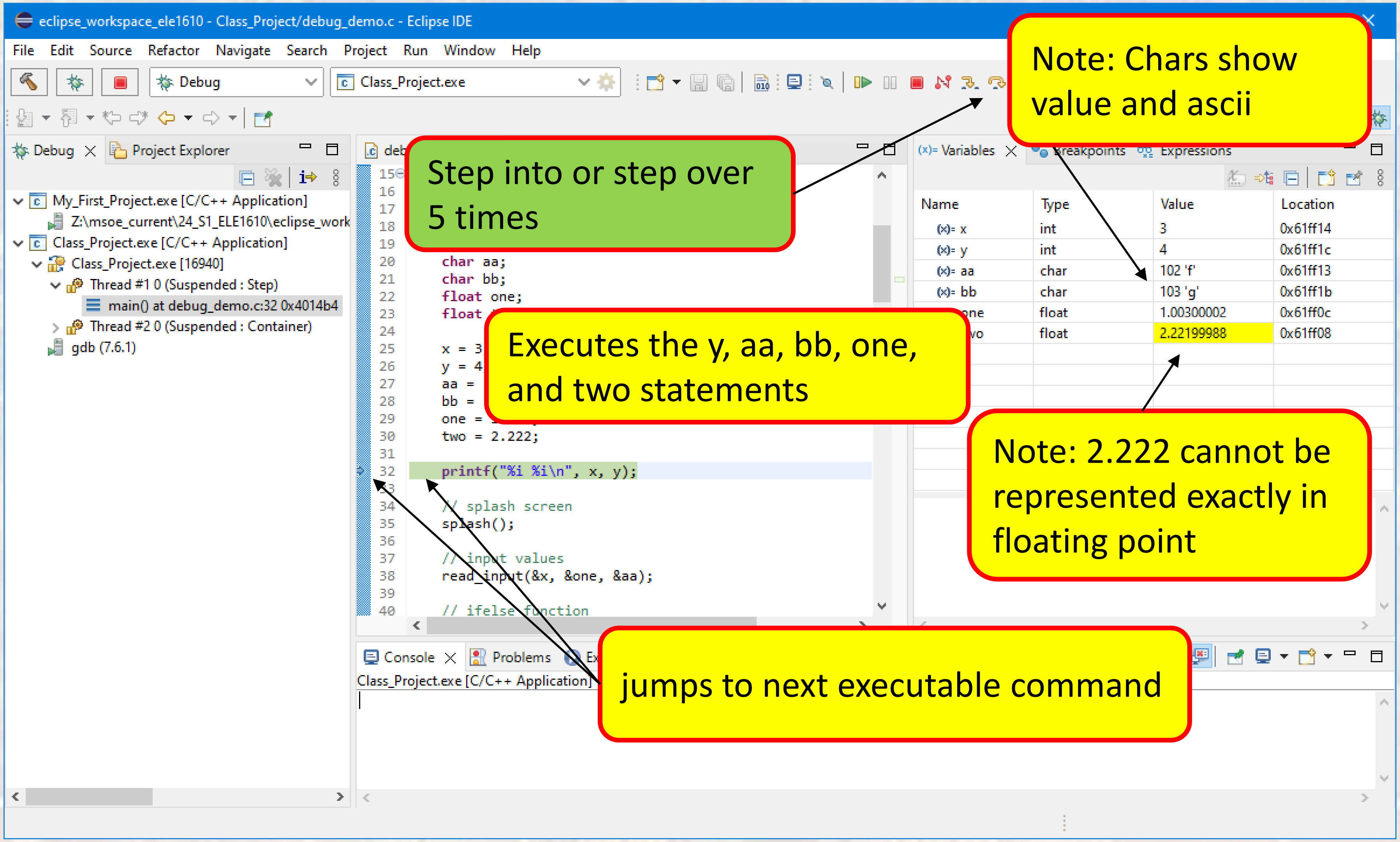
Step into or step over

Executes the x=3 statement

Value change highlighted in yellow

jumps to next executable command

Debug



eclipse_workspace_ele1610 - Class_Project/debug_demo.c - Eclipse IDE

```
My_First_Project.exe [C/C++ Application]
    Z:\msoe_current\24_S1_ELE1610\eclipse_work
Class_Project.exe [C/C++ Application]
  Class_Project.exe [16940]
    Thread #1 0 (Suspended : Step)
      main() at debug_demo.c:32 0x4014b4
    Thread #2 0 (Suspended : Container)
  gdb (7.6.1)
```

```
    char aa;
    char bb;
    float one;
    float 

    x = 3
    y = 4
    aa = 
    bb = 
    one = 
    two = 2.222;

    printf("%i %i\n", x, y);

    // splash screen
    splash();

    // input values
    read_input(&x, &one, &aa);

    // ifelse function
```

Name	Type	Value	Location
x	int	3	0x61ff14
y	int	4	0x61ff1c
aa	char	102 'f'	0x61ff13
bb	char	103 'g'	0x61ff1b
one	float	1.00300002	0x61ff0c
two	float	2.22199988	0x61ff08

Class_Project.exe [C/C++ Application]

Note: Chars show value and ascii

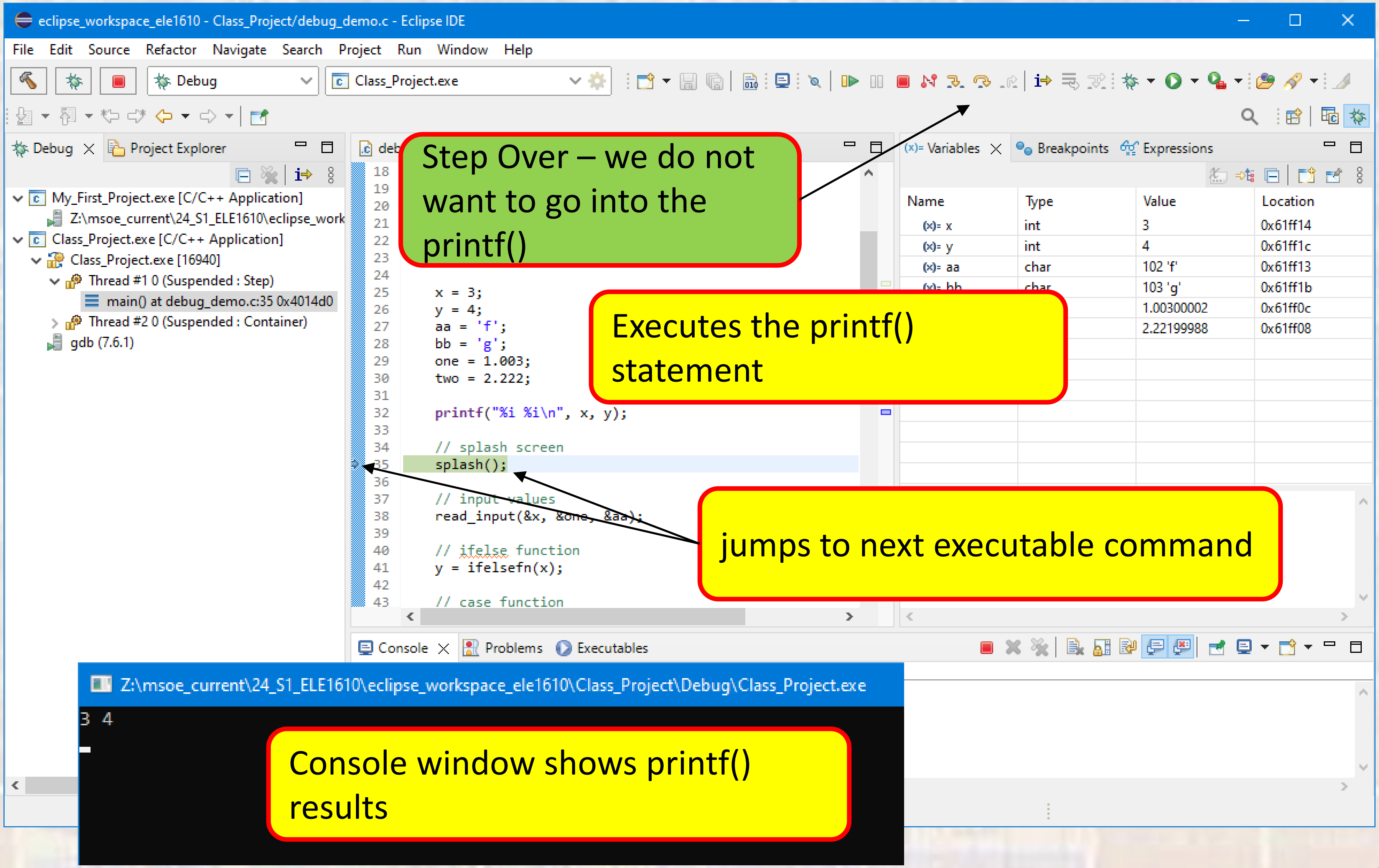
Step into or step over 5 times

Executes the y, aa, bb, one, and two statements

Note: 2.222 cannot be represented exactly in floating point

jumps to next executable command

Debug



Step Over – we do not want to go into the printf()

Executes the printf() statement

jumps to next executable command

Console window shows printf() results

```
x = 3;
y = 4;
aa = 'f';
bb = 'g';
one = 1.003;
two = 2.222;

printf("%i %i\n", x, y);

// splash screen
splash();

// input values
read_input(&x, &one, &aa);

// ifelse function
y = ifelsefn(x);

// case function
```

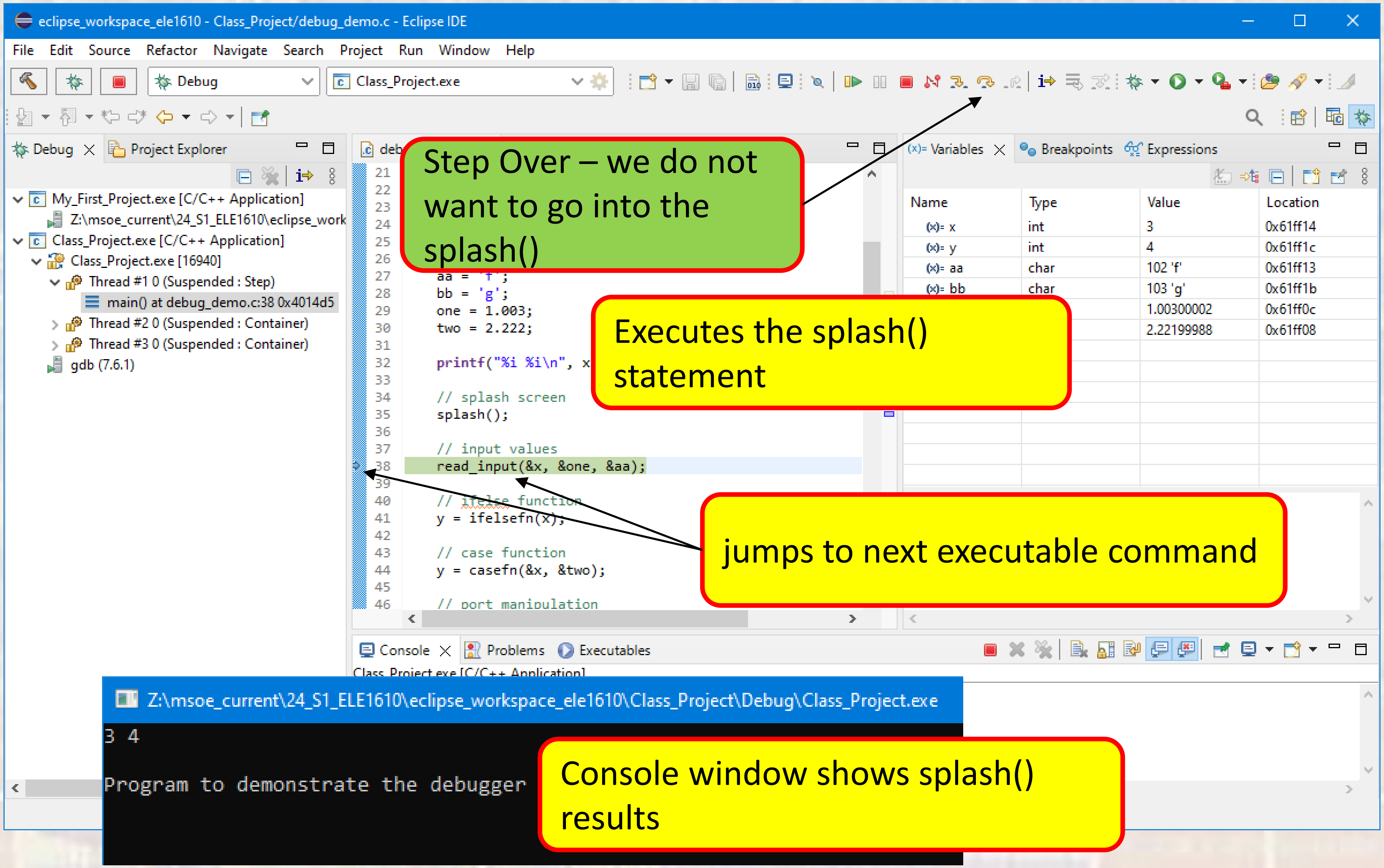
Name	Type	Value	Location
x	int	3	0x61ff14
y	int	4	0x61ff1c
aa	char	102 'f'	0x61ff13
bb	char	103 'g'	0x61ff1b
		1.00300002	0x61ff0c
		2.22199988	0x61ff08

Thread #1 0 (Suspended : Step)
main() at debug_demo.c:35 0x4014d0

Z:\msoe_current\24_S1_ELE1610\eclipse_workspace_ele1610\Class_Project\Debug\Class_Project.exe

```
3 4
```

Debug



eclipse_workspace_ele1610 - Class_Project/debug_demo.c - Eclipse IDE

```
My_First_Project.exe [C/C++ Application]
  Z:\msoe_current\24_S1_ELE1610\eclipse_work
Class_Project.exe [C/C++ Application]
  Class_Project.exe [16940]
    Thread #1 0 (Suspended : Step)
      main() at debug_demo.c:38 0x4014d5
    Thread #2 0 (Suspended : Container)
    Thread #3 0 (Suspended : Container)
  gdb (7.6.1)
```

```
    aa = 'f';
    bb = 'g';
    one = 1.003;
    two = 2.222;

    printf("%i %i\n", x

    // splash screen
    splash();

    // input values
    read_input(&x, &one, &aa);

    // ifelse function
    y = ifelsefn(x);

    // case function
    y = casefn(&x, &two);

    // port manipulation
```

Name	Type	Value	Location
(x)= x	int	3	0x61ff14
(x)= y	int	4	0x61ff1c
(x)= aa	char	102 'f'	0x61ff13
(x)= bb	char	103 'g'	0x61ff1b
		1.00300002	0x61ff0c
		2.22199988	0x61ff08

Class_Project.exe [C/C++ Application]

Z:\msoe_current\24_S1_ELE1610\eclipse_workspace_ele1610\Class_Project\Debug\Class_Project.exe

```
3 4

Program to demonstrate the debugger
```

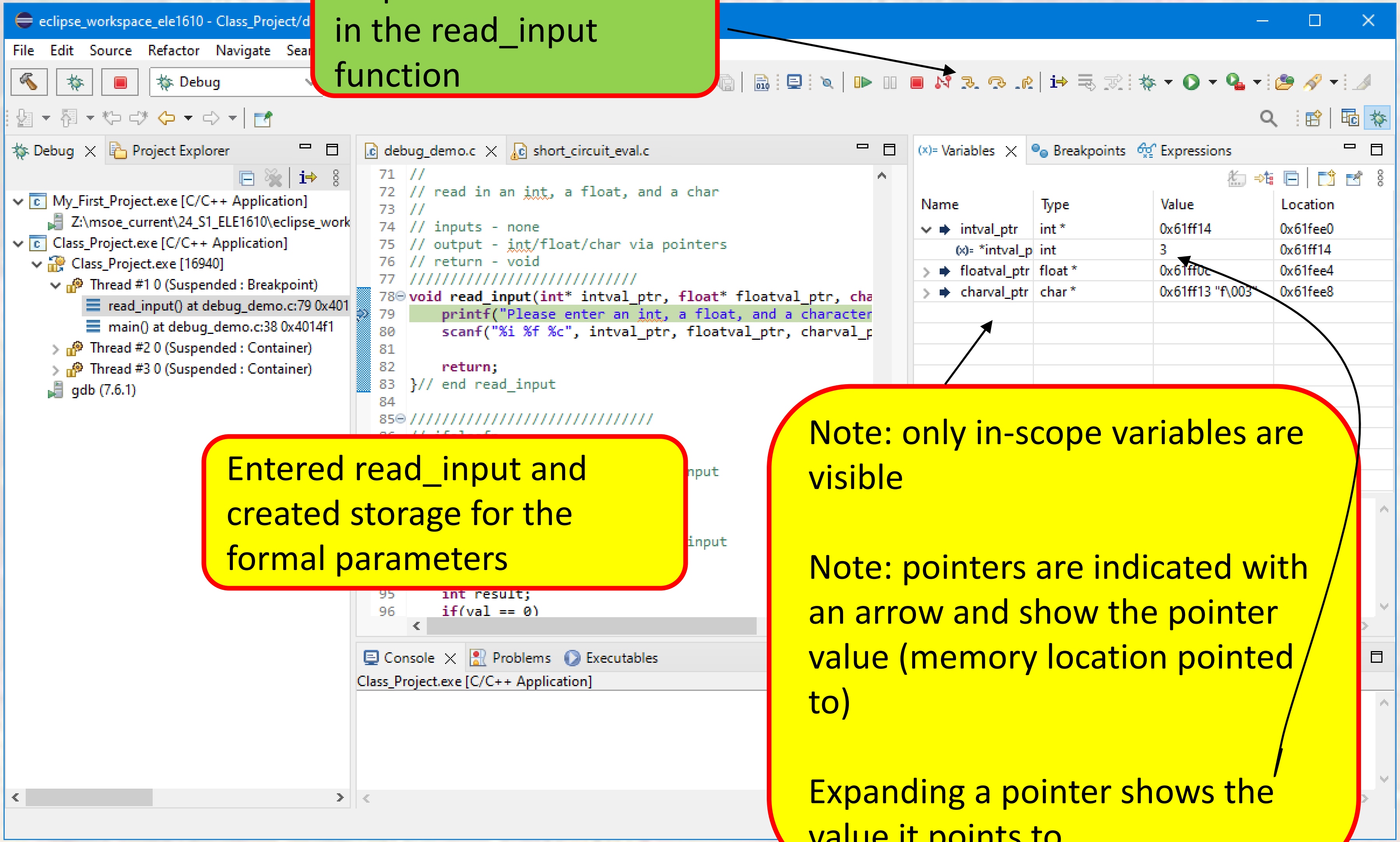
Step Over – we do not want to go into the splash()

Executes the splash() statement

jumps to next executable command

Console window shows splash() results

Debug



Step Into – we are now in the read_input function

Entered read_input and created storage for the formal parameters

Note: only in-scope variables are visible

Note: pointers are indicated with an arrow and show the pointer value (memory location pointed to)

Expanding a pointer shows the value it points to

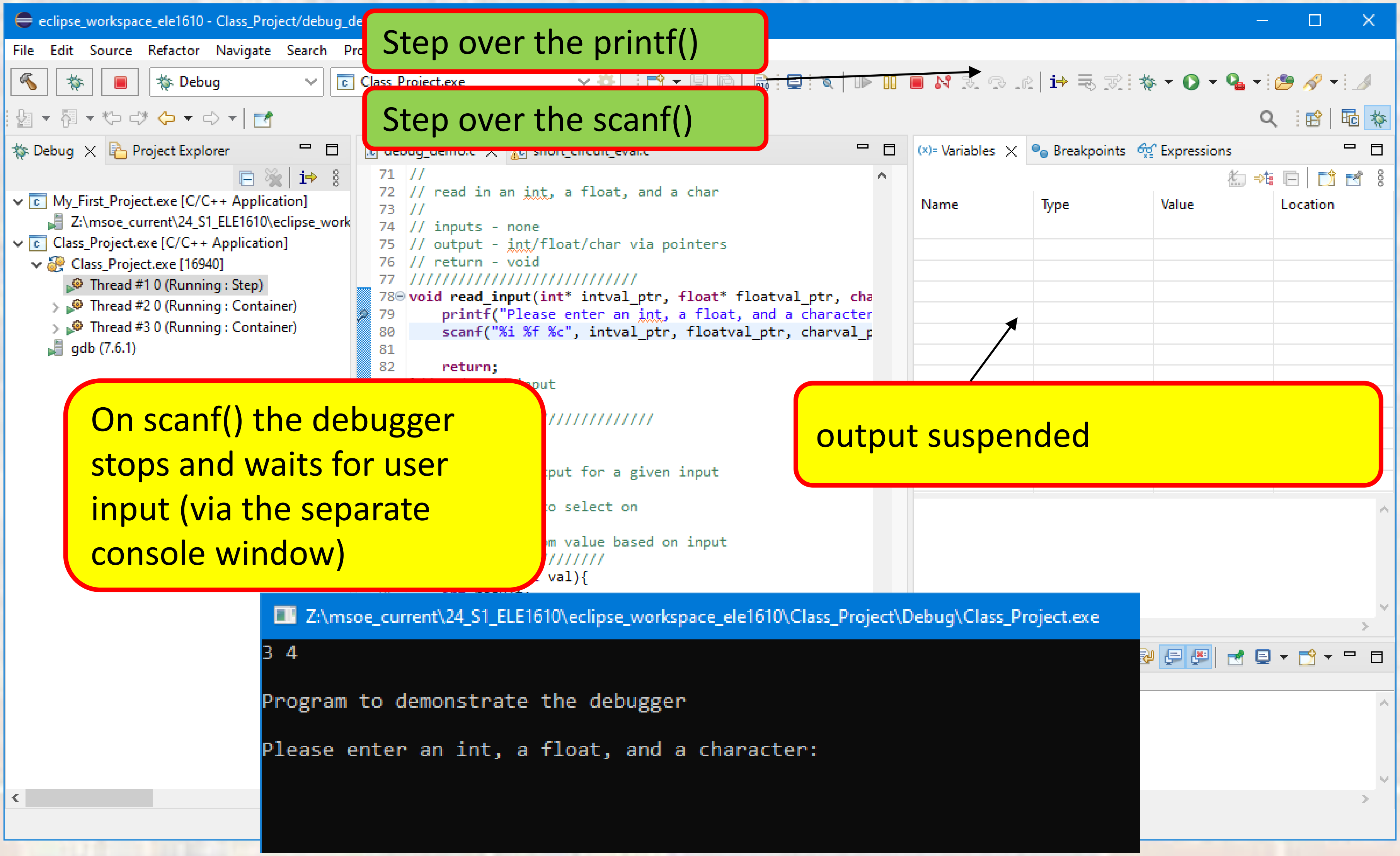
```
My_First_Project.exe [C/C++ Application]
  Z:\msoe_current\24_S1_ELE1610\eclipse_work
Class_Project.exe [C/C++ Application]
  Class_Project.exe [16940]
    Thread #1 0 (Suspended : Breakpoint)
      read_input() at debug_demo.c:79 0x401
      main() at debug_demo.c:38 0x4014f1
    Thread #2 0 (Suspended : Container)
    Thread #3 0 (Suspended : Container)
  gdb (7.6.1)
```

```
//
// read in an int, a float, and a char
//
// inputs - none
// output - int/float/char via pointers
// return - void
/////////////////////////////
void read_input(int* intval_ptr, float* floatval_ptr, cha
    printf("Please enter an int, a float, and a character
    scanf("%i %f %c", intval_ptr, floatval_ptr, charval_p

    return;
}// end read_input

/////////////////////////////
```

Name	Type	Value	Location
intval_ptr	int *	0x61ff14	0x61fee0
*intval_p	int	3	0x61ff14
floatval_ptr	float *	0x61ff0c	0x61fee4
charval_ptr	char *	0x61ff13 "f\003"	0x61fee8

Class_Project.exe [C/C++ Application]

Debug


eclipse_workspace_ele1610 - Class_Project/debug_de
File Edit Source Refactor Navigate Search Pro
Debug
Class_Project.exe
Debug
Project Explorer
Variables
Breakpoints
Expressions
Name
Type
Value
Location

Step over the printf()

Step over the scanf()

```
My_First_Project.exe [C/C++ Application]
    Z:\msoe_current\24_S1_ELE1610\eclipse_work
Class_Project.exe [C/C++ Application]
  Class_Project.exe [16940]
    Thread #1 0 (Running : Step)
    Thread #2 0 (Running : Container)
    Thread #3 0 (Running : Container)
    gdb (7.6.1)
```

```
//
// read in an int, a float, and a char
//
// inputs - none
// output - int/float/char via pointers
// return - void
/////////////////////////////
void read_input(int* intval_ptr, float* floatval_ptr, cha
    printf("Please enter an int, a float, and a character
    scanf("%i %f %c", intval_ptr, floatval_ptr, charval_p

    return;
```

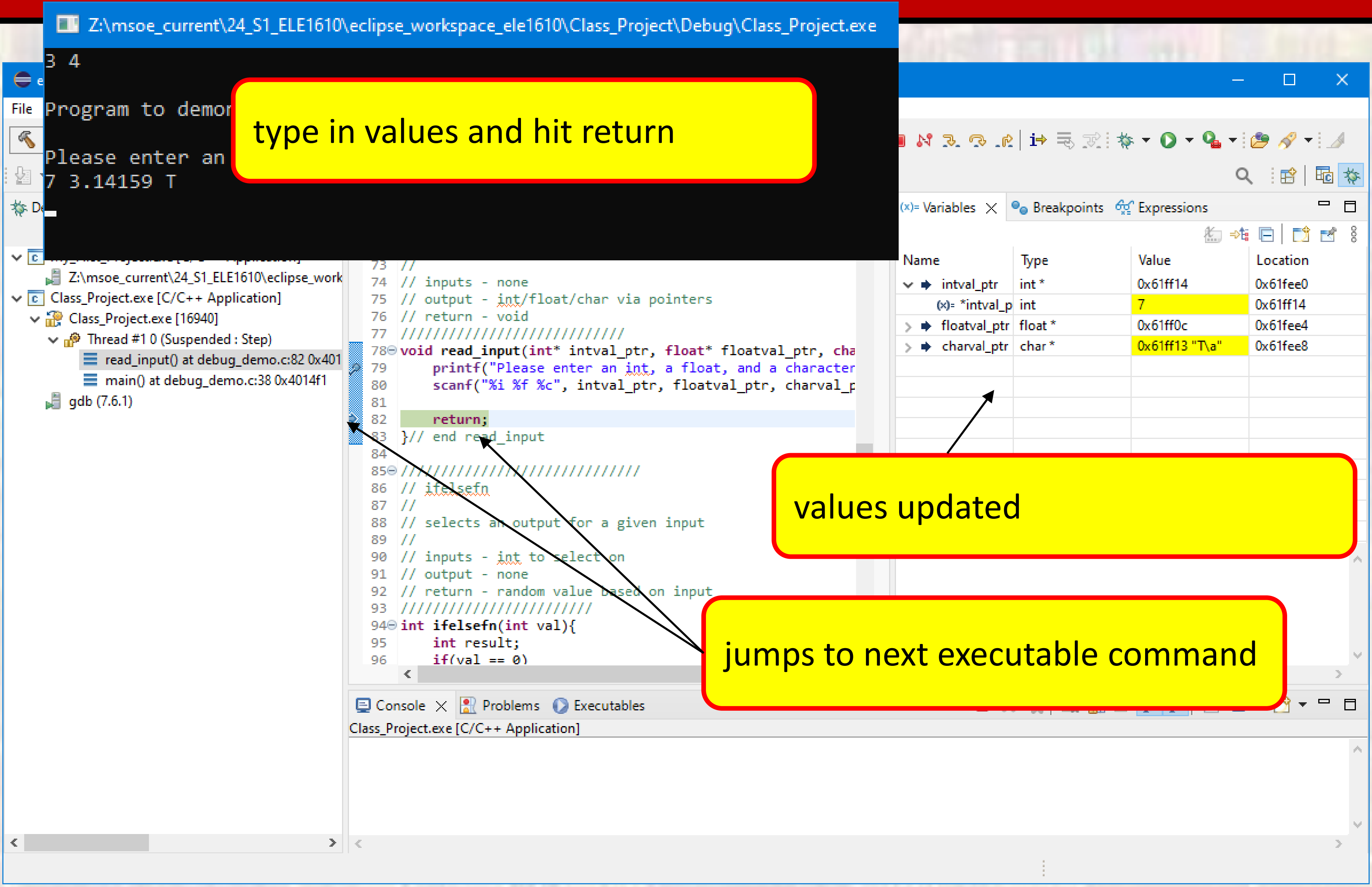
On scanf() the debugger stops and waits for user input (via the separate console window)

output suspended

Z:\msoe_current\24_S1_ELE1610\eclipse_workspace_ele1610\Class_Project\Debug\Class_Project.exe

```
3 4

Program to demonstrate the debugger

Please enter an int, a float, and a character:
```

Debug



Z:\msoe_current\24_S1_ELE1610\eclipse_workspace_ele1610\Class_Project\Debug\Class_Project.exe

```
3 4

Program to demor

Please enter an
7 3.14159 T
```

type in values and hit return

```
Z:\msoe_current\24_S1_ELE1610\eclipse_work
Class_Project.exe [C/C++ Application]
  Class_Project.exe [16940]
    Thread #1 0 (Suspended : Step)
      read_input() at debug_demo.c:82 0x401
      main() at debug_demo.c:38 0x4014f1
  gdb (7.6.1)
```

```
//
// inputs - none
// output - int/float/char via pointers
// return - void
////////////////////////////
void read_input(int* intval_ptr, float* floatval_ptr, cha
    printf("Please enter an int, a float, and a character
    scanf("%i %f %c", intval_ptr, floatval_ptr, charval_p

    return;
}// end read_input

/////////////////////////////
// ifelsefn
//
// selects an output for a given input
//
// inputs - int to select on
// output - none
// return - random value based on input
//////////////////////
int ifelsefn(int val){
    int result;
    if(val == 0)
```

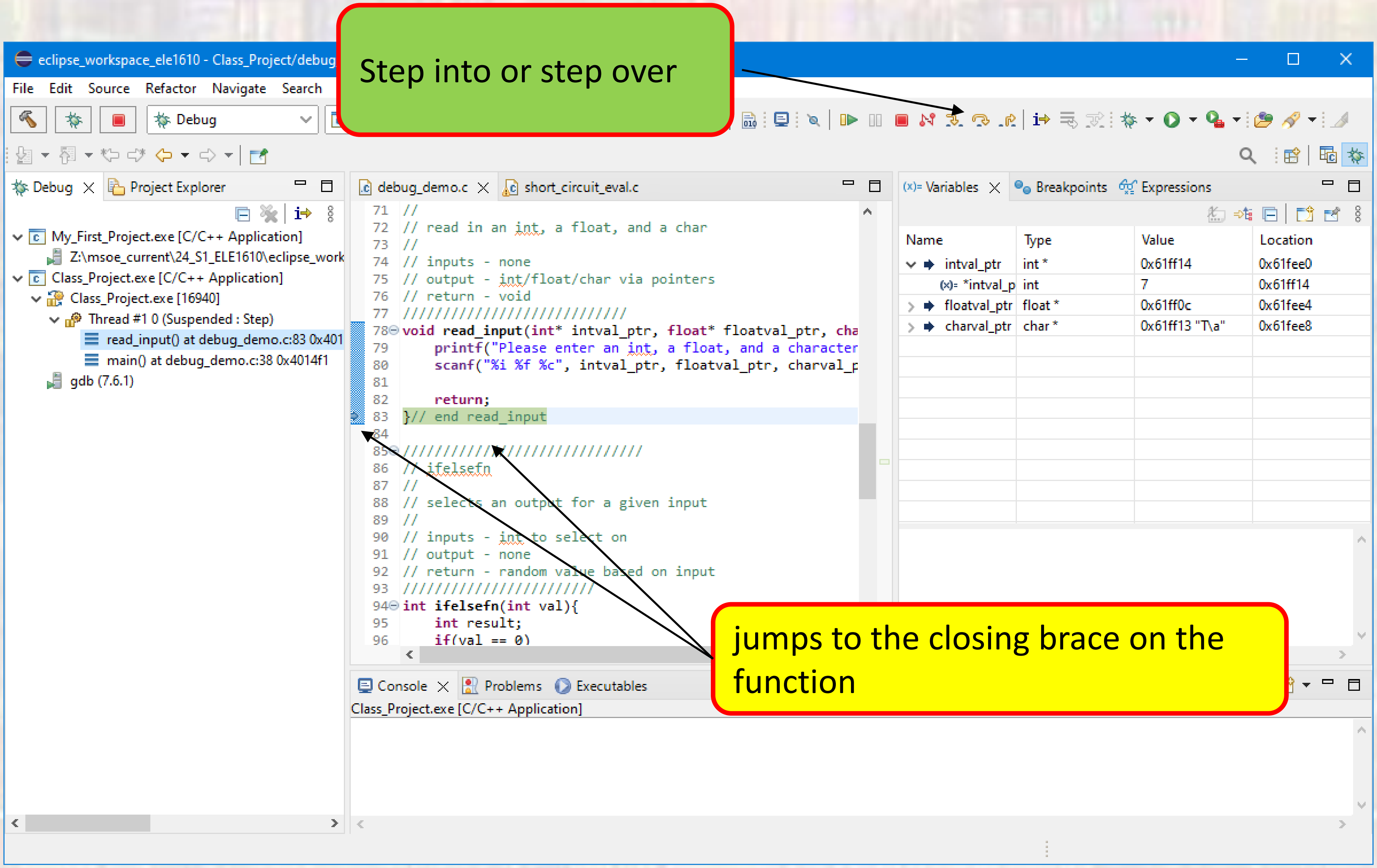
Name	Type	Value	Location
intval_ptr	int *	0x61ff14	0x61fee0
*intval_p	int	7	0x61ff14
floatval_ptr	float *	0x61ff0c	0x61fee4
charval_ptr	char *	0x61ff13 "T\a"	0x61fee8

values updated

jumps to next executable command

Class_Project.exe [C/C++ Application]

Debug



Step into or step over

jumps to the closing brace on the function

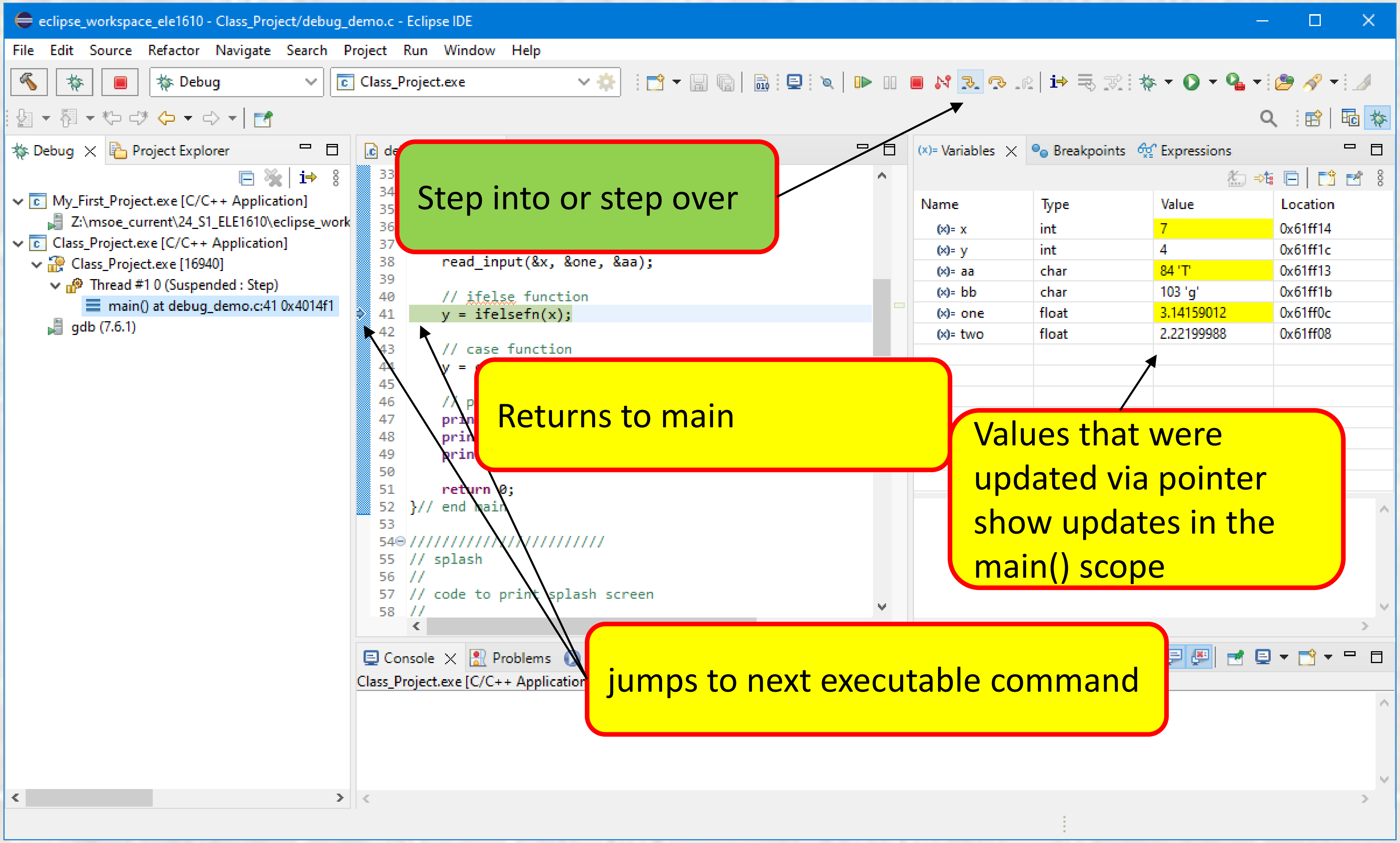
```
//
// read in an int, a float, and a char
//
// inputs - none
// output - int/float/char via pointers
// return - void
/////////////////////////////
void read_input(int* intval_ptr, float* floatval_ptr, cha
    printf("Please enter an int, a float, and a character
    scanf("%i %f %c", intval_ptr, floatval_ptr, charval_p

    return;
}// end read_input

//////////////////////////////
// ifelsefn
//
// selects an output for a given input
//
// inputs - int to select on
// output - none
// return - random value based on input
/////////////////////////////
int ifelsefn(int val){
    int result;
    if(val == 0)
```

Debug:
- My_First_Project.exe [C/C++ Application]
  - Z:\msoe_current\24_S1_ELE1610\eclipse_work
- Class_Project.exe [C/C++ Application]
  - Class_Project.exe [16940]
    - Thread #1 0 (Suspended : Step)
      - read_input() at debug_demo.c:83 0x401
      - main() at debug_demo.c:38 0x4014f1
  - gdb (7.6.1)

Name	Type	Value	Location
intval_ptr	int *	0x61ff14	0x61fee0
*intval_p	int	7	0x61ff14
floatval_ptr	float *	0x61ff0c	0x61fee4
charval_ptr	char *	0x61ff13 "T\a"	0x61fee8

Console: Class_Project.exe [C/C++ Application]

Debug


eclipse_workspace_ele1610 - Class_Project/debug_demo.c - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Class_Project.exe
Debug
Project Explorer
Variables
Breakpoints
Expressions
Console
Problems

```
My_First_Project.exe [C/C++ Application]
    Z:\msoe_current\24_S1_ELE1610\eclipse_work
Class_Project.exe [C/C++ Application]
    Class_Project.exe [16940]
        Thread #1 0 (Suspended : Step)
            main() at debug_demo.c:41 0x4014f1
    gdb (7.6.1)
```

```
    read_input(&x, &one, &aa);

    // ifelse function
    y = ifelsefn(x);

    // case function
    y =

    // p
    prin
    prin
    prin

    return 0;
}// end main

////////////////////////
// splash
//
// code to print splash screen
//
```

Name	Type	Value	Location
x	int	7	0x61ff14
y	int	4	0x61ff1c
aa	char	84 'T'	0x61ff13
bb	char	103 'g'	0x61ff1b
one	float	3.14159012	0x61ff0c
two	float	2.22199988	0x61ff08

Class_Project.exe [C/C++ Application]

Step into or step over

Returns to main

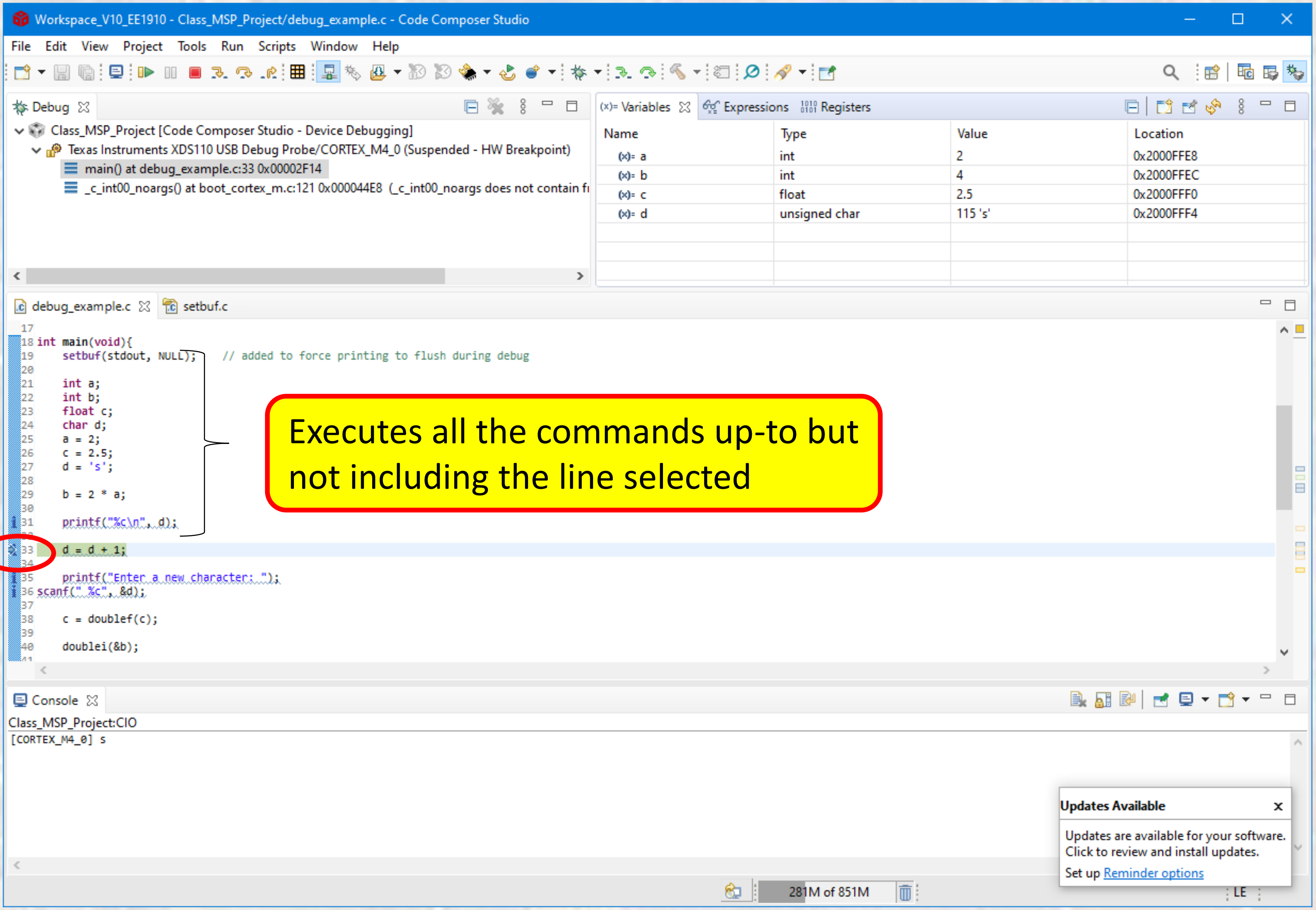
Values that were updated via pointer show updates in the main() scope

jumps to next executable command

Additional Things we can do in the debugger

Debug

Instead of single stepping
Right click on a line and select run_to



Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio

Class_MSP_Project [Code Composer Studio - Device Debugging]
Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended - HW Breakpoint)
main() at debug_example.c:33 0x00002F14
_c_int00_noargs() at boot_cortex_m.c:121 0x000044E8 (_c_int00_noargs does not contain f

Name	Type	Value	Location
a	int	2	0x2000FFE8
b	int	4	0x2000FFEC
c	float	2.5	0x2000FFF0
d	unsigned char	115 's'	0x2000FFF4

```
int main(void){
    setbuf(stdout, NULL);     // added to force printing to flush during debug

    int a;
    int b;
    float c;
    char d;
    a = 2;
    c = 2.5;
    d = 's';

    b = 2 * a;

    printf("%c\n", d);

    d = d + 1;

    printf("Enter a new character: ");
scanf(" %c", &d);

    c = doublef(c);

    doublei(&b);
```

Executes all the commands up-to but not including the line selected

Class_MSP_Project:CIO

```
[CORTEX_M4_0] s
```

Updates Available

Updates are available for your software. Click to review and install updates.

Set up Reminder options

281M of 851M

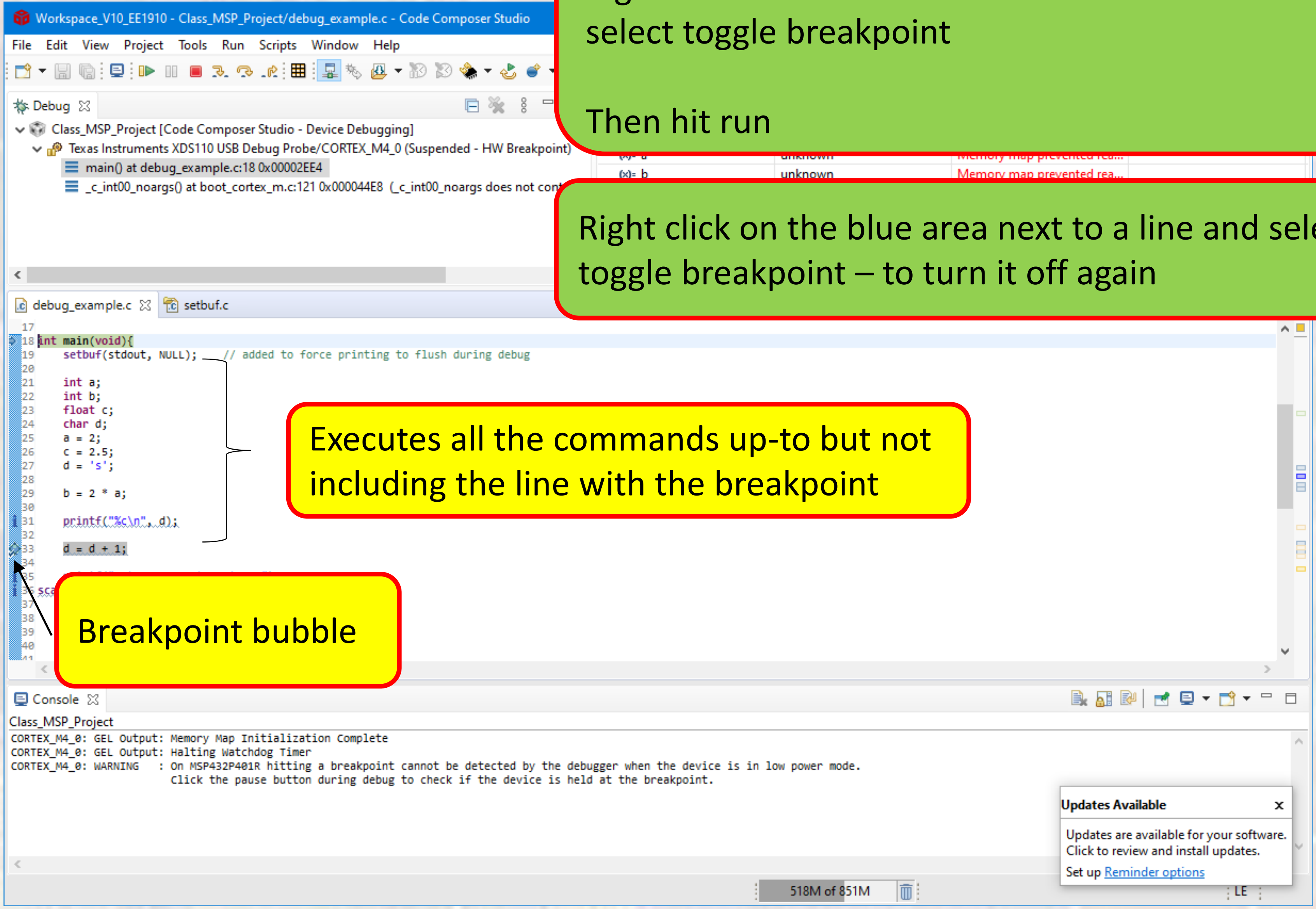
Debug

Instead of single stepping

Right click on the blue area next to a line and select toggle breakpoint

Then hit run

Right click on the blue area next to a line and select toggle breakpoint – to turn it off again


Workspace_V10_EE1910 - Class_MSP_Project/debug_example.c - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
Debug
debug_example.c
setbuf.c
Console
(x)= b
unknown
Memory map prevented rea...
518M of 851M
LE

Class_MSP_Project [Code Composer Studio - Device Debugging]
Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended - HW Breakpoint)
main() at debug_example.c:18 0x00002EE4
_c_int00_noargs() at boot_cortex_m.c:121 0x000044E8 (_c_int00_noargs does not con

```
int main(void){
    setbuf(stdout, NULL);   // added to force printing to flush during debug

    int a;
    int b;
    float c;
    char d;
    a = 2;
    c = 2.5;
    d = 's';

    b = 2 * a;

    printf("%c\n", d);

    d = d + 1;


sca
```

Executes all the commands up-to but not including the line with the breakpoint

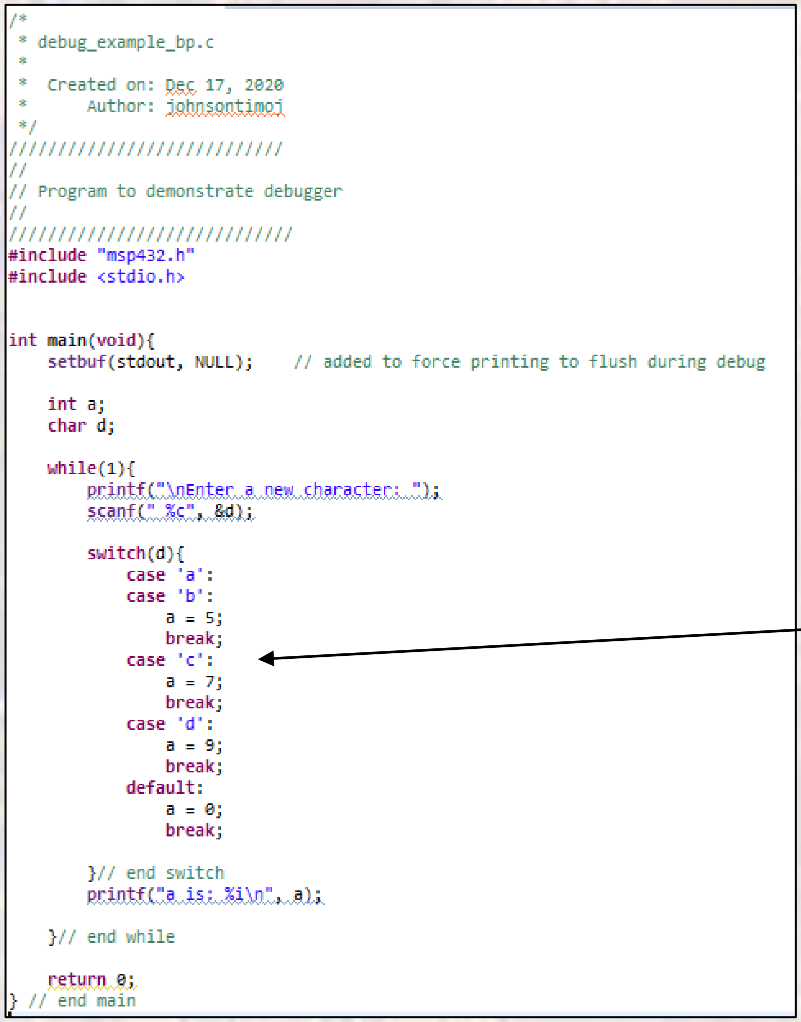
Breakpoint bubble

Class_MSP_Project

```
CORTEX_M4_0: GEL Output: Memory Map Initialization Complete
CORTEX_M4_0: GEL Output: Halting Watchdog Timer
CORTEX_M4_0: WARNING   : On MSP432P401R hitting a breakpoint cannot be detected by the debugger when the device is in low power mode.
                         Click the pause button during debug to check if the device is held at the breakpoint.
```

Updates Available
Updates are available for your software. Click to review and install updates.
Set up Reminder options

Debug



```
/*
 * debug_example_bp.c
 *
 *  Created on: Dec 17, 2020
 *      Author: johnsontimoj
 */
////////////////////////////
//
// Program to demonstrate debugger
//
/////////////////////////////
#include "msp432.h"
#include <stdio.h>


int main(void){
    setbuf(stdout, NULL);    // added to force printing to flush during debug

    int a;
    char d;

    while(1){
        printf("\nEnter a new character: ");
        scanf(" %c", &d);

        switch(d){
            case 'a':
            case 'b':
                a = 5;
                break;
            case 'c':
                a = 7;
                break;
            case 'd':
                a = 9;
                break;
            default:
                a = 0;
                break;

        }// end switch
        printf("a is: %i\n", a);

    }// end while

    return 0;
} // end main
```

Would like to stop when a c is entered

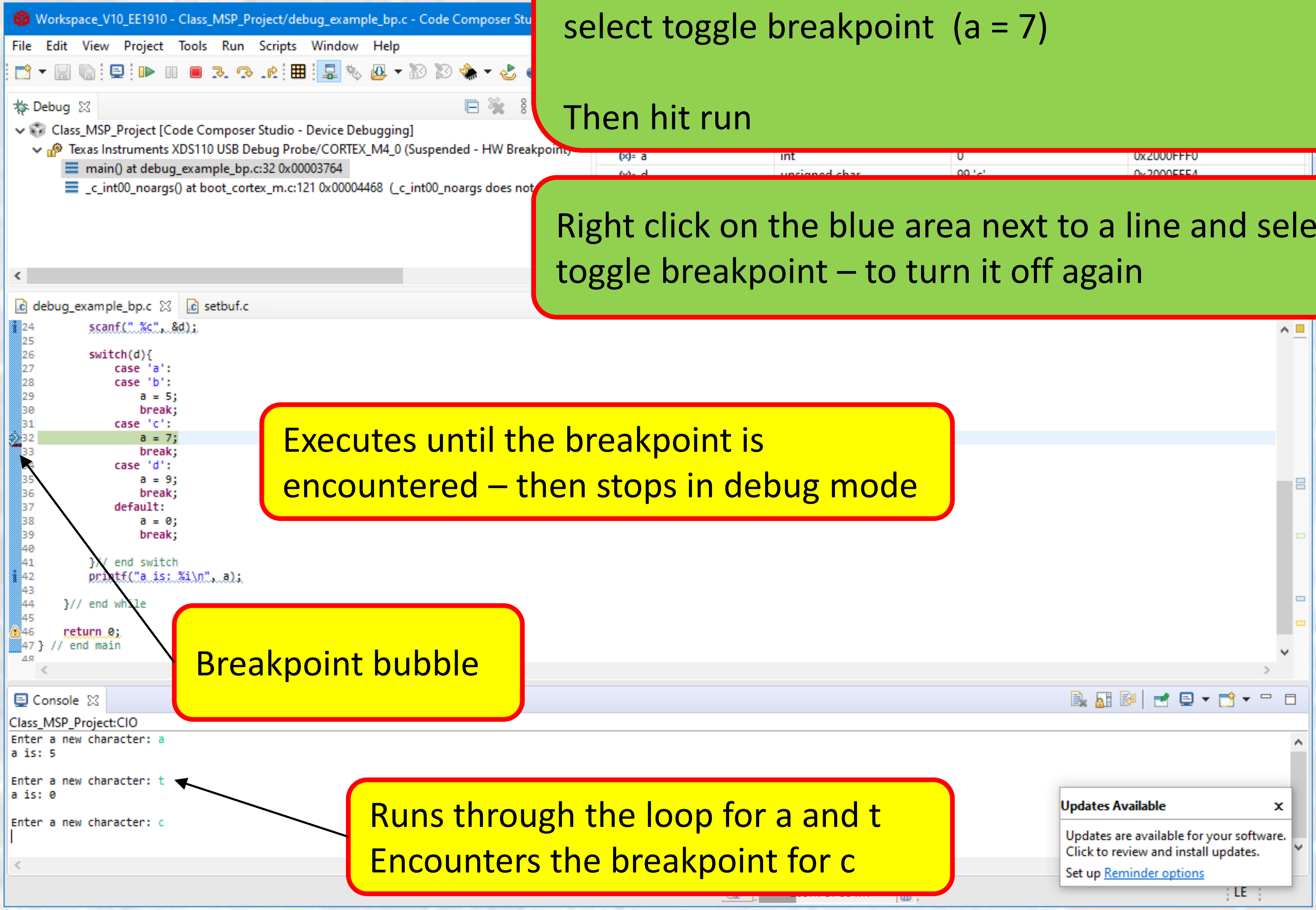
Debug

Instead of single stepping

Right click on the blue area next to a line and select toggle breakpoint (a = 7)

Then hit run

Right click on the blue area next to a line and select toggle breakpoint – to turn it off again


Workspace_V10_EE1910 - Class_MSP_Project/debug_example_bp.c - Code Composer Stu
File Edit View Project Tools Run Scripts Window Help
Debug
debug_example_bp.c
setbuf.c
Console
Updates Available
Set up Reminder options

```
Class_MSP_Project [Code Composer Studio - Device Debugging]
  Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended - HW Breakpoint)
    main() at debug_example_bp.c:32 0x00003764
    _c_int00_noargs() at boot_cortex_m.c:121 0x00004468 (_c_int00_noargs does not
```

```
(x)= a      int             0         0x2000FFF0
(x)= d      unsigned char   99 'c'    0x2000FFF4
```

```
        scanf(" %c", &d);

        switch(d){
            case 'a':
            case 'b':
                a = 5;
                break;
            case 'c':
                a = 7;
                break;
            case 'd':
                a = 9;
                break;
            default:
                a = 0;
                break;

        }// end switch
        printf("a is: %i\n", a);

    }// end while

    return 0;
} // end main
```

```
Class_MSP_Project:CIO
Enter a new character: a
a is: 5

Enter a new character: t
a is: 0

Enter a new character: c
```

Updates are available for your software. Click to review and install updates.

Executes until the breakpoint is encountered – then stops in debug mode

Breakpoint bubble

Runs through the loop for a and t
Encounters the breakpoint for c