



Week 2 - Lecture 1

Control Statements – Selection

Edited by: Dr. Wooi Ping Cheah
Autumn 2022

Overview

- Write a C Program
- Compile and Run a C Program
- Language Insecurities
- Practice Hygienic Coding

Program Structure

- All C programs require a main function.

```
2 void main(void)
3 {
4
5 }
```

- Without it ...

```
c:/mingw/bin/../../lib/gcc/mingw32/5.3.0/../../../../libmingw32.a(main.o):(.text.startup+0xa0):
undefined reference to `WinMain@16'
collect2.exe: error: ld returned 1 exit status
```

- Too many of it ...

```
C:\Users\z2017233\Desktop>gcc lecture2.c -o lecture2
lecture2.c:8:6: error: redefinition of 'main'
void main(void)
^
lecture2.c:2:6: note: previous definition of 'main' was here
void main(void)
^
```

Program Structure (2)

- C programs contain one or more functions, one of which *must* be `main`.
- Every program in C begins executing at the function `main`.
- The keyword `void/int` to the left of `main` indicates that `main` “returns” nothing/an integer (whole number) value.

Program Structure (3)

Program starts at the beginning of main. A left brace, {, begins the **body of code**, whereas a corresponding **right brace** ends.

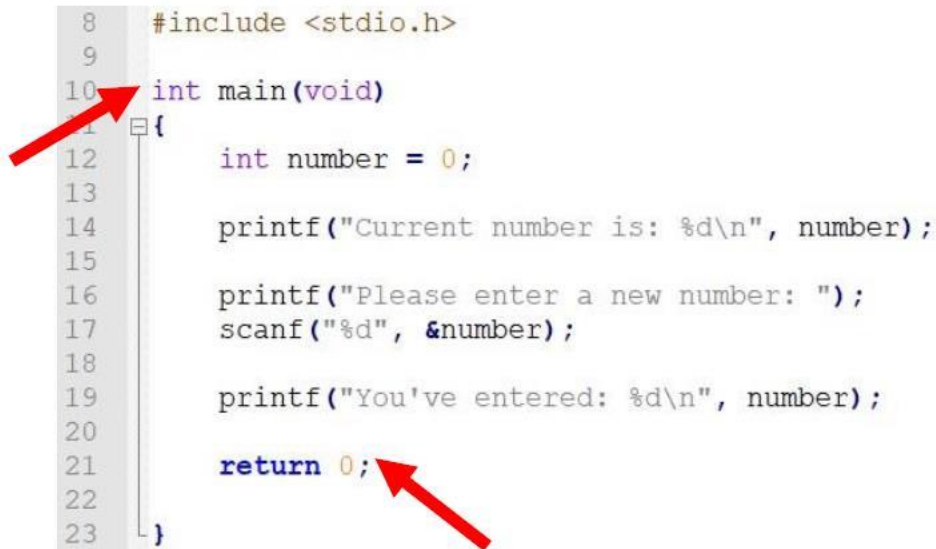
```
8  #include <stdio.h>
9
10 int main(void)
11 {
12     int number = 0;
13
14     printf("Current number is: %d\n", number);
15
16     printf("Please enter a new number: ");
17     scanf("%d", &number);
18
19     printf("You've entered: %d\n", number);
20
21     return 0;
22 }
23
```

- This pair of braces and the portion of the program between the braces is called a block

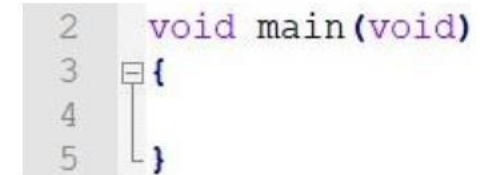
Return Value

- It is common to return 0 to indicate that the program has run and exited successfully.

```
8  #include <stdio.h>
9
10 int main(void)
11 {
12     int number = 0;
13
14     printf("Current number is: %d\n", number);
15
16     printf("Please enter a new number: ");
17     scanf("%d", &number);
18
19     printf("You've entered: %d\n", number);
20
21     return 0;
22 }
23
```



```
2  void main(void)
3  {
4
5  }
```



Return Value (2)

- A program can have multiple functions.
- Each function may or may not return a value.

```
26 #include <stdio.h>
27
28 void myPrint(void);
29 int myReturn(void);
30
31 int main(void)
32 {
33     int number = 0;
34
35     myPrint();
36
37     printf("Current number is: %d\n", number);
38
39     number = myReturn();
40
41     printf("The number is now: %d\n", number);
42
43     return 0;
44 }
45
46 void myPrint(void)
47 {
48     printf("Hello There !!\n");
49 }
50
51 int myReturn(void)
52 {
53     return 5;
54 }
55
```

```
C:\Users\z2017233\Desktop>lecture2
Hello There!!
Current number is: 0
The number is now: 5
C:\Users\z2017233\Desktop>_
```

Libraries

- Not all programs need libraries.


- Example libraries:

- limits.h
- math.h
- stdio.h

- Without it ...

```
2 void main(void)
3 {
4
5 }
```

```
8 #include <stdio.h>
9
10 int main(void)
11 {
12     int number = 0;
13
14     printf("Current number is: %d\n", number);
15
16     printf("Please enter a new number: ");
17     scanf("%d", &number);
18
19     printf("You've entered: %d\n", number);
20
21     return 0;
22 }
23
```

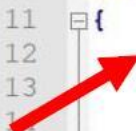


```
C:\Users\z2017233\Desktop>gcc lecture2.c -o lecture2
lecture2.c: In function 'main':
lecture2.c:13:2: warning: implicit declaration of function 'printf' [-Wimplicit-function-declaration]
  printf("Current number is: %d", number);
  ^
lecture2.c:13:2: warning: incompatible implicit declaration of built-in function 'printf'
lecture2.c:13:2: note: include '<stdio.h>' or provide a declaration of 'printf'
lecture2.c:16:2: warning: implicit declaration of function 'scanf' [-Wimplicit-function-declaration]
  scanf("%d", &number);
  ^
lecture2.c:16:2: warning: incompatible implicit declaration of built-in function 'scanf'
lecture2.c:16:2: note: include '<stdio.h>' or provide a declaration of 'scanf'
```


Variables and Data Types

- Variables must be declared before first use.

```
8  #include <stdio.h>
9
10 int main(void)
11 {
12     int number = 0;
13
14     printf("Current number is: %d\n", number);
15
16     printf("Please enter a new number: ");
17     scanf("%d", &number);
18
19     printf("You've entered: %d\n", number);
20
21     return 0;
22 }
23
```



- Otherwise ...

```
C:\Users\z2017233\Desktop>gcc lecture2.c -o lecture2
lecture2.c: In function 'main':
lecture2.c:31:36: error: 'number' undeclared (first use in this function)
  printf("Current number is: %d\n", number);
                                   ^
lecture2.c:31:36: note: each undeclared identifier is reported only once for each function it appears in
```

Variables and Data Types (2)

- Variables can be initialised.


```
26 #include <stdio.h>
27
28 int main(void)
29 {
30     int counter;
31     int number = 0;
32
33     while(number < 1)
34     {
35         printf("Please enter a new number: ");
36         scanf("%d", &number);
37
38         counter = counter + 1;
39     }
40
41     if(counter > 1)
42     {
43         printf("User has entered %d new numbers\n", counter);
44     }
45     else
46     {
47         printf("User has entered %d new number\n", counter);
48     }
49
50     return 0;
51 }
52
```

```
C:\Users\z2017233\Desktop>gcc lecture2.c -o lecture2
C:\Users\z2017233\Desktop>lecture2
Please enter a new number: 1
User has entered 4194433 new numbers
```

Input with scanf

- Read data from the standard input stream (stdin) and store that data in variables

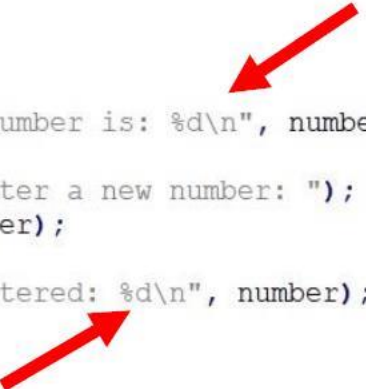
```
8  #include <stdio.h>
9
10 int main(void)
11 {
12     int number = 0;
13
14     printf("Current number is: %d\n", number);
15
16     printf("Please enter a new number: ");
17     scanf("%d", &number);
18
19     printf("You've entered: %d\n", number);
20
21     return 0;
22 }
23
```



Output with printf

- Output “**formatted**” data to the standard output e.g. monitor.

```
8  #include <stdio.h>
9
10 int main(void)
11 {
12     int number = 0;
13
14     printf("Current number is: %d\n", number);
15
16     printf("Please enter a new number: ");
17     scanf("%d", &number);
18
19     printf("You've entered: %d\n", number);
20
21     return 0;
22 }
23
```



- Using correct format specifier is important!!

Some useful characters for Printf()

Escape sequence	Description
<code>\n</code>	Newline. Position the cursor at the beginning of the next line.
<code>\t</code>	Horizontal tab. Move the cursor to the next tab stop.
<code>\a</code>	Alert. Produces a sound or visible alert without changing the current cursor position.
<code>\\</code>	Backslash. Insert a backslash character in a string.
<code>\"</code>	Double quote. Insert a double-quote character in a string.

Comments

- Use block or single line comment to explain what your program does.

```
149  int main(void)
150  {
151      /* This program calculate the remainder if division,
152         and return zero to the shell */
153
154      int i = (10 % 3);
155
156      // The line belows return zero to the shell that calls the program
157      return 0;
158  }
```

Overview

- Write a C Program
- **Compile and Run a C Program**
- Language Insecurities
- Practice Hygienic Coding

Compile a C Program

- gcc filename.c -o filename

Compile and link
i.e. reference to functions
e.g. standard libraries

Create an executable called “filename”

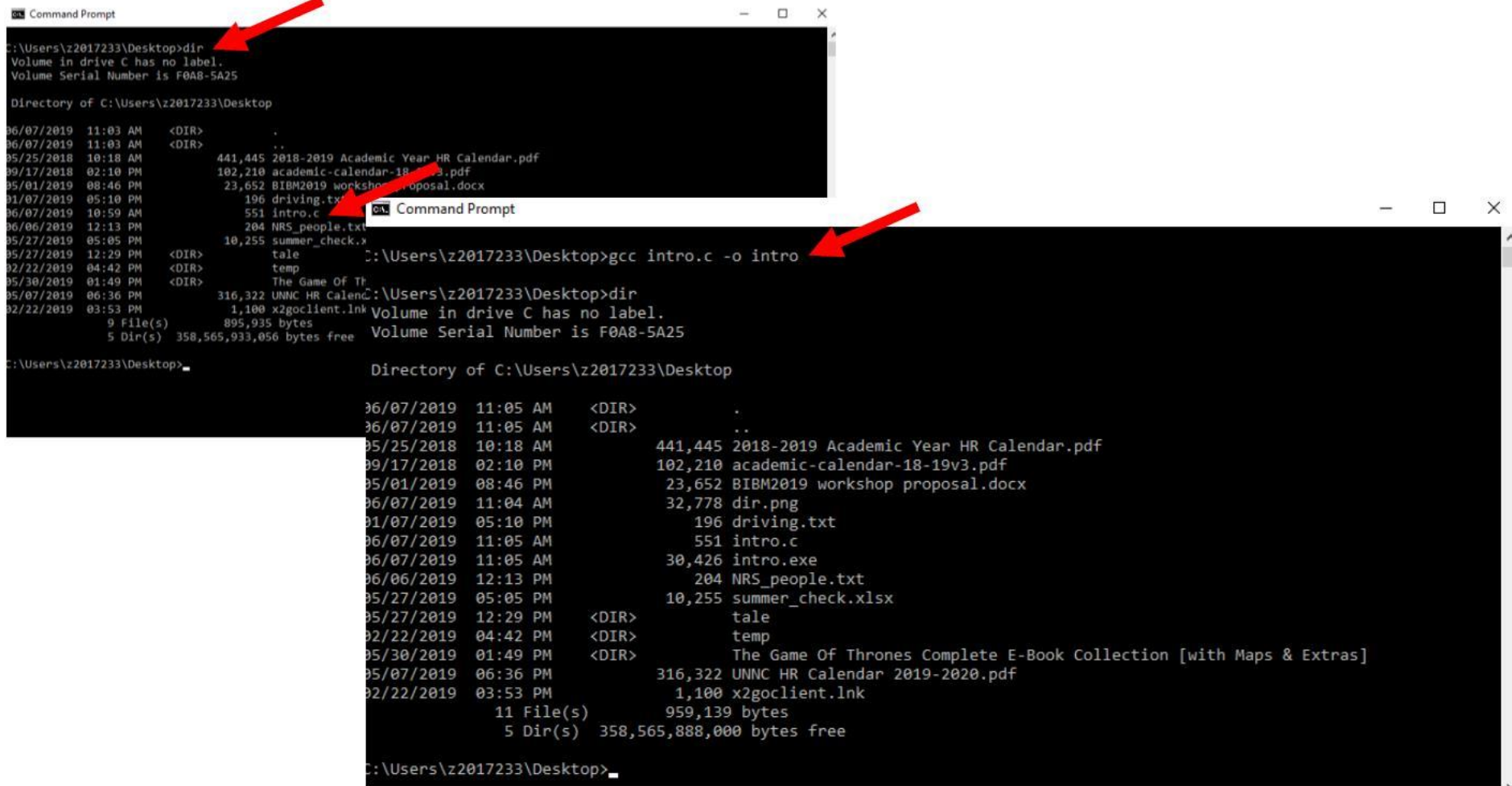
DO NOT type in filename.c !!!!!

```
C:\Users\z2017233\Desktop>gcc selection.c -o selection.c  
gcc: fatal error: input file 'selection.c' is the same as output file  
compilation terminated.
```

- gcc filename.c

Create “default” an executable call a.out

Compile a C Program - Windows



```
Command Prompt
C:\Users\z2017233\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is F0A8-5A25

Directory of C:\Users\z2017233\Desktop

06/07/2019  11:03 AM  <DIR>          .
06/07/2019  11:03 AM  <DIR>          ..
05/25/2018  10:18 AM             441,445 2018-2019 Academic Year HR Calendar.pdf
09/17/2018  02:10 PM             102,210 academic-calendar-18-19v3.pdf
05/01/2019  08:46 PM              23,652 BIBM2019 workshop proposal.docx
01/07/2019  05:10 PM              196 driving.txt
06/07/2019  10:59 AM              551 intro.c
06/06/2019  12:13 PM              204 NRS_people.txt
05/27/2019  05:05 PM             10,255 summer_check.xls
05/27/2019  12:29 PM  <DIR>          tale
02/22/2019  04:42 PM  <DIR>          temp
05/30/2019  01:49 PM  <DIR>          The Game Of Th
05/07/2019  06:36 PM             316,322 UNNC HR Calend
02/22/2019  03:53 PM              1,100 x2goclient.lnk
          9 File(s)      895,935 bytes
          5 Dir(s)  358,565,933,056 bytes free

C:\Users\z2017233\Desktop>

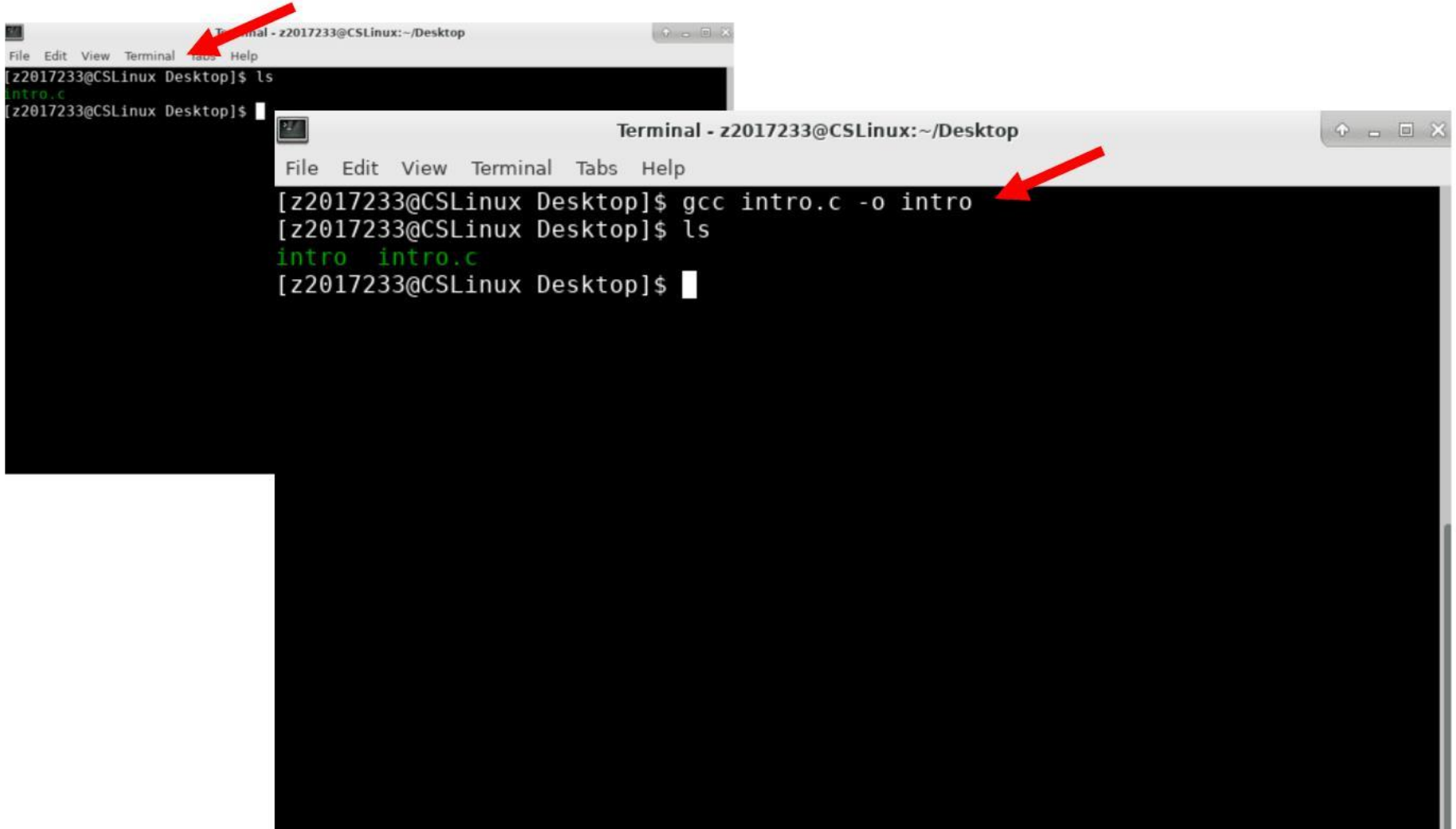
Command Prompt
C:\Users\z2017233\Desktop>gcc intro.c -o intro
C:\Users\z2017233\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is F0A8-5A25

Directory of C:\Users\z2017233\Desktop

06/07/2019  11:05 AM  <DIR>          .
06/07/2019  11:05 AM  <DIR>          ..
05/25/2018  10:18 AM             441,445 2018-2019 Academic Year HR Calendar.pdf
09/17/2018  02:10 PM             102,210 academic-calendar-18-19v3.pdf
05/01/2019  08:46 PM              23,652 BIBM2019 workshop proposal.docx
06/07/2019  11:04 AM             32,778 dir.png
01/07/2019  05:10 PM              196 driving.txt
06/07/2019  11:05 AM              551 intro.c
06/07/2019  11:05 AM             30,426 intro.exe
06/06/2019  12:13 PM              204 NRS_people.txt
05/27/2019  05:05 PM             10,255 summer_check.xlsx
05/27/2019  12:29 PM  <DIR>          tale
02/22/2019  04:42 PM  <DIR>          temp
05/30/2019  01:49 PM  <DIR>          The Game Of Thrones Complete E-Book Collection [with Maps & Extras]
05/07/2019  06:36 PM             316,322 UNNC HR Calendar 2019-2020.pdf
02/22/2019  03:53 PM              1,100 x2goclient.lnk
          11 File(s)      959,139 bytes
          5 Dir(s)  358,565,888,000 bytes free

C:\Users\z2017233\Desktop>
```

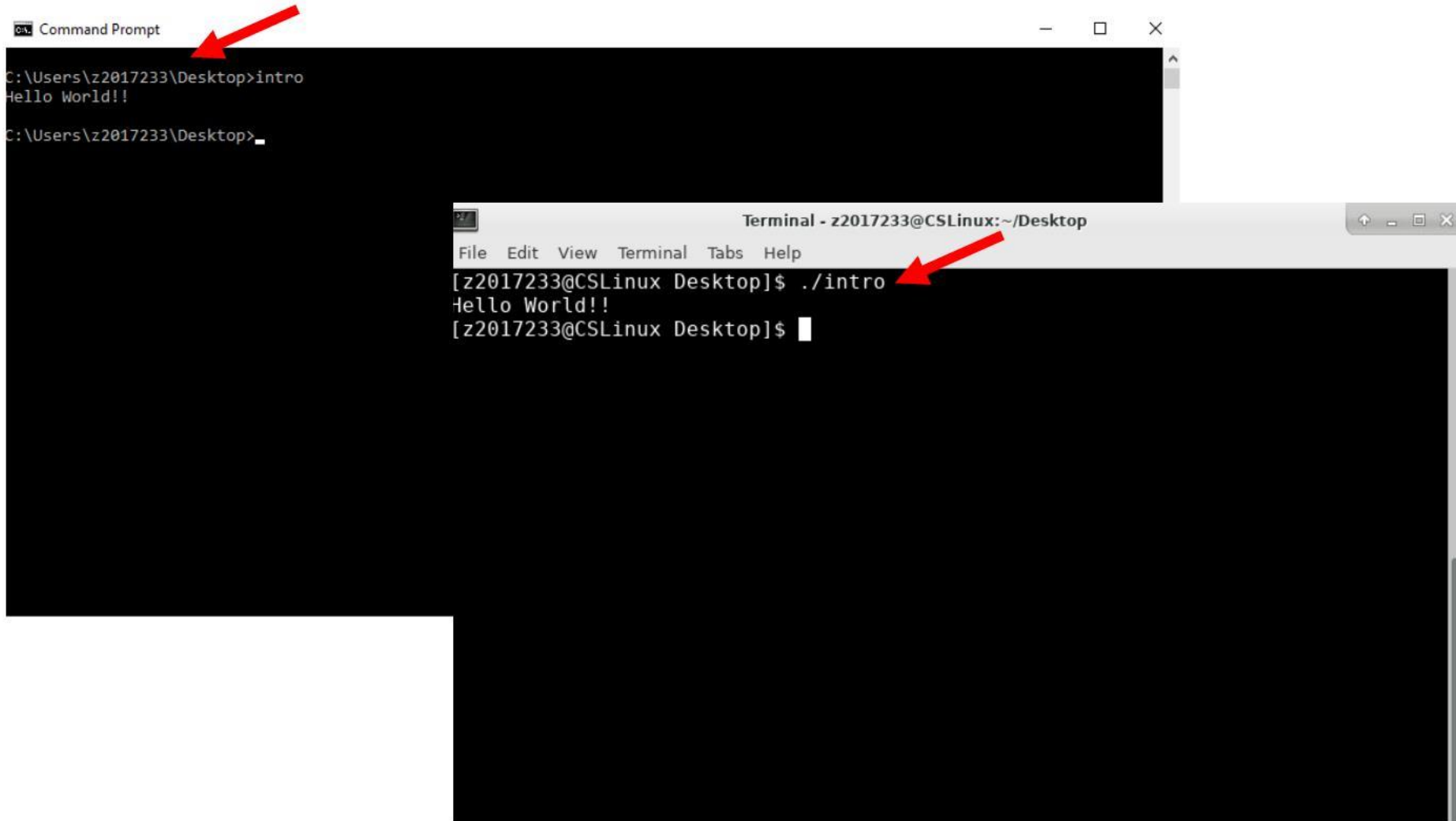
Compile a C Program - Linux



```
Terminal - z2017233@CSLinux: ~/Desktop
File Edit View Terminal Tabs Help
[z2017233@CSLinux Desktop]$ ls
intro.c
[z2017233@CSLinux Desktop]$

Terminal - z2017233@CSLinux: ~/Desktop
File Edit View Terminal Tabs Help
[z2017233@CSLinux Desktop]$ gcc intro.c -o intro
[z2017233@CSLinux Desktop]$ ls
intro intro.c
[z2017233@CSLinux Desktop]$
```

Run a C Program



The image displays two terminal windows side-by-side. The top window is a Windows Command Prompt titled 'Command Prompt' with a red arrow pointing to the title bar. It shows the command 'intro' being executed at the path 'C:\Users\z2017233\Desktop', resulting in the output 'Hello World!!'. The bottom window is a Linux Terminal titled 'Terminal - z2017233@CSLinux:~/Desktop' with a red arrow pointing to the command prompt. It shows the command './intro' being executed, also resulting in the output 'Hello World!!'.

```
Command Prompt
C:\Users\z2017233\Desktop>intro
Hello World!!
C:\Users\z2017233\Desktop>_

Terminal - z2017233@CSLinux:~/Desktop
File Edit View Terminal Tabs Help
[z2017233@CSLinux Desktop]$ ./intro
Hello World!!
[z2017233@CSLinux Desktop]$
```

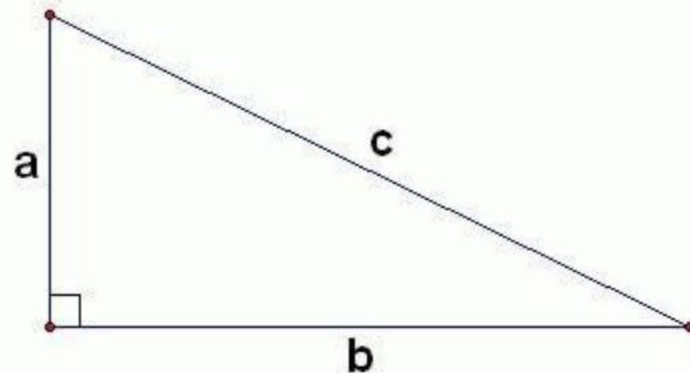
Example 1: Sum

```
1 // Fig. 2.5: fig02_05.c
2 // Addition program.
3 #include <stdio.h>
4
5 // function main begins program execution
6 int main( void )
7 {
8     int integer1; // first number to be entered by user
9     int integer2; // second number to be entered by user
10    int sum; // variable in which sum will be stored
11
12    printf( "Enter first integer\n" ); // prompt
13    scanf( "%d", &integer1 ); // read an integer
14
15    printf( "Enter second integer\n" ); // prompt
16    scanf( "%d", &integer2 ); // read an integer
17
18    sum = integer1 + integer2; // assign total to sum
19
20    printf( "Sum is %d\n", sum ); // print sum
21 }
```

Example 2: Right-Angled Triangle

```
143 #include <stdio.h>
144 #include <stdlib.h>
145
146 int main(int argc, char *argv[])
147 {
148     int x, y, z;
149
150     printf("Enter value for x: ");
151     scanf("%d", &x);
152     if(x < 1)
153     {
154         printf("Invalid value\n");
155         exit(1);
156     }
157
158     printf("Enter value for y: ");
159     scanf("%d", &y);
160     if(y < 1)
161     {
162         printf("Invalid value\n");
163         exit(1);
164     }
165
166     printf("Enter value for z: ");
167     scanf("%d", &z);
168     if(z < 1)
169     {
170         printf("Invalid value\n");
171         exit(1);
172     }
```

```
174     int lhs = x * x + y * y;
175     int rhs = z * z;
176
177     if(lhs == rhs)
178     {
179         printf("Right angled triangle\n");
180     }
181     else
182     {
183         printf("Not right angled, %d does not equal %d\n", lhs, rhs );
184     }
185 }
186 }
```



$$a^2 + b^2 = c^2$$

Source: <https://mathblog.com/reference/theorems/pythagorean-theorem/>



Overview

- Write a C Program
- Compile and Run a C Program
- **Language Insecurities**
- Practice Hygienic Coding

Language Insecurities

- Style and expressiveness
- Valid typos
- Error detection
- Misunderstanding the language
- Wrong expectations
- Run-time error detection

Style and Expressiveness

- How clearly the language constructs can "express" the developer's intentions.
- For example, switch statement cases must end with break, return, or a comment indicating a fall-through

```
1 #include <stdio.h>
2
3 void doSomething();
4 void doSomethingElse();
5 void doDefaultThing();
6
7 int main()
8 {
9     int value = 0;
10
11     switch(value)
12     {
13         case 1:
14             doSomething();
15
16         case 2:
17             doSomethingElse();
18             break;
19
20         default:
21             doDefaultThing();
22             break;
23     }
24 }
```

```
1 #include <stdio.h>
2
3 void doSomething();
4 void doSomethingElse();
5 void doDefaultThing();
6
7 int main()
8 {
9     int value = 0;
10
11     switch(value)
12     {
13         case 1:
14             doSomething();
15             /* falls through */
16
17         case 2:
18             doSomethingElse();
19             break;
20
21         default:
22             doDefaultThing();
23             break;
24     }
25 }
```


Valid Typos

```
1 #include <stdio.h>
2 int main()
3 {
4     int a = 0;
5     if(a = 1)
6         printf("a is NOT equal to zero\n");
7     else
8         printf("a is equal to zero\n");
9
10    return 0;
11 }
```

```
1 #include <stdio.h>
2 int main()
3 {
4     int a = 0;
5     if(a == 1)
6         printf("a is NOT equal to zero\n");
7     else
8         printf("a is equal to zero\n");
9
10    return 0;
11 }
```

Error Detection

```
1 #include <stdio.h>
2 int main()
3 {
4     int b = 1.25;
5     double c = 1.25;
6
7     printf("The sum of b and c is %.2f\n", (c+b));
8
9     return 0;
10 }
```



Understanding the Language

```
1 #include <stdio.h>
2 int main()
3 {
4     int d = 1;
5     int e = 2;
6     int f = 3;
7     printf("%d\n", (d+e*f));
8     printf("%d\n", ((d+e)*f));
9
10    return 0;
11 }
```

Wrong Expectations

- Be careful when you copy and paste code.


```
113  #include <stdio.h>
114
115  int main(void)
116  {
117      printf("Hello World!!\n");
118      Printf("Hello World!!\n");
119      return 0;
120
121 }
```

```
C:\Users\z2017233\Desktop>gcc lecture2.c -o lecture2
lecture2.c: In function 'main':
lecture2.c:118:2: warning: implicit declaration of function 'Printf' [-Wimplicit-function-declaration]
  Printf(â?oHello World!!\nâ??);
   ^
lecture2.c:118:2: error: stray '\342' in program
lecture2.c:118:2: error: stray '\200' in program
lecture2.c:118:2: error: stray '\234' in program
lecture2.c:118:12: error: 'Hello' undeclared (first use in this function)
  Printf(â?oHello World!!\nâ??);
   ^
lecture2.c:118:12: note: each undeclared identifier is reported only once for each function it appears in
lecture2.c:118:18: error: expected ')' before 'World'
  Printf(â?oHello World!!\nâ??);
   ^
lecture2.c:118:18: error: stray '\' in program
lecture2.c:118:18: error: stray '\342' in program
lecture2.c:118:18: error: stray '\200' in program
lecture2.c:118:18: error: stray '\235' in program
C:\Users\z2017233\Desktop>
```

Run-time Error Detection

- Array out-of-bound is not detected.

```
113  #include <stdio.h>
114
115  int main(void)
116  {
117      int arr[2];
118      arr[0] = 0;
119      arr[1] = 1;
120
121      int i = 0;
122      for(i = 0; i < 3; i++)
123      {
124          printf("%d\n", arr[i]);
125      }
126
127      return 0;
128
129  }
```



Overview

- Write a C Program
- Compile and Run a C Program
- Language Insecurities
- **Practice Hygienic Coding**

Hygienic Coding

- All variables, pointers and references are properly initialised at first and subsequent uses
- All input data, messages and output data should be validated
- Implementations of all algorithms should be validated
- Error handling
- Resource access are explicitly managed
- Use of comment statements
- Code layout and use of indenting
- Layout of braces “{ }” and block structures
- Statement complexity

Summary

- Write a C Program
- Compile and Run a C Program
- Language Insecurities
- Practice Hygienic Coding

