

## Lectures/Week\_10/week10\_tutorial.c

```
1 #include <stdio.h> // Standard Input Output library
2 #include <stdlib.h> // Standard library for general utilities
3
4 // Main function: Entry point of the program
5 // Every C program must have a main function. It's where the program starts.
6 int main()
7 {
8     /* 03/11's Content: C Programming Basics, Arrays
9     =====
10    */
11    printf("03/11's Content: C Programming Basics, Arrays\n");
12    printf("=====\\n");
13
14    // printf is a function in <stdio.h> that prints output to the console.
15    // The "\\n" is a newline character that moves the cursor to the next line.
16    // The text inside the double quotes is the message to be printed.
17    printf("Hello, world!\\n");
18
19    // Semicolon (;) is used to terminate a statement in C.
20    // It tells the compiler that the statement is complete.
21    // Unlike MATLAB where semicolon (;) is optional.
22    // In C, every statement must end with a semicolon.
23
24    // Print an additional message
25    printf("Welcome to learning C programming!\\n");
26
27    // Declare and initialize variables
28    int a = 5;
29    int b = 10;
30    int sum = a + b;
31
32    // int is a data type for integers
33    // Variables a, b, and sum are declared as integers and initialized.
34    // Unlike MATLAB, C requires explicit declaration of variable types.
35    // In C, you must declare the type of each variable before using it.
36
37    // int a = 5; // first declaration must include type
38    // a = 6; // further assignment does not require type
39    // so int a=6 here again would be an error
40
41    // Print the sum of two numbers
42    printf("The sum of %d and %d is %d\\n", a, b, sum);
43
44    int M[10] = {1, 2, 3, 40, 5, 60, 7, 80, 9, 101}; // Array of integers
45    // you need to specify the size of the array when you declare it, using []
46    // the initial value of an array is included inside a curly braces {}
```

```

47 // the elements of the array are separated by commas
48 // Note the type of M is int, so the type of each element is also int
49
50 printf("the fourth element of M is %d\n", M[3]); // index of C starts from 0
51 // KEY DIFFERENCE FROM MATLAB
52 // In C, the index of an array starts from 0, not 1.
53
54 int N[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}; // 2D array of integers
55
56 printf("the element at row 1 and column 3 of N is %d\n", N[0][2]);
57 printf("\n\n");
58
59 // Return 0 to indicate successful execution
60 // The main function (declared as int function) must return an int value
61 // back to the operating system.
62 // Returning 0 typically means that the program executed successfully.
63
64 /* 03/13's Content: Data Types, Printf, Scnaf, Precision
65 =====
66 */
67 printf("03/13's Content: Data Types, Printf, Scnaf, Precision\n");
68 printf("=====\\n");
69
70 // Data Types
71 // C has several data types, including int, float, double, char, etc.
72 int x = 10, d = 100; // integer
73 float y = 3.14, e = 2.72; // single-precision floating-point number
74 double z = 3.14159, f = 1.23456; // double-precision floating-point number
75 char ch = 'A', option = '1'; // character
76
77 // Printf
78 printf("x=%d\\n", x); // %d is a placeholder for an integer
79 printf("y=%f\\n", y); // %f is a placeholder for a float
80 printf("z=%lf\\n", z); // %lf is a placeholder for a double
81 printf("ch=%c\\n", ch); // %c is a placeholder for a character
82 printf("option=%c\\n", option); // single quotes of anything is a character
83
84 // Never mess up the placeholders and the variables types
85 // Although it may not cause an error, it may lead to undefined results
86 printf("z in integer is %d\\n", z); // %d is for a float, but z is an double
87 printf("ch in integer is %d\\n", ch); // %d is for an int, but ch is a
character
88
89 // Scnaf
90 // scanf is a function in <stdio.h> that reads input from the console
91 printf("Enter an integer for x: ");
92 scanf("%d", &x); // Read an integer from the user and store it in x
93 // It would directly modify x

```

```

94 // scanf("%d", &m); // illegal, m is not declared
95 // scanf("%d", x); // illegal, & is a must when using scanf
96 // scanf("%d", &x, &y); // illegal, num of placeholder mismatched
97 // scanf("%f", &x); // illegal, type mismatched
98 printf("new x=%d\n", x); // Print the value of x, should be new now!
99
100 printf("Enter a double for z: ");
101 scanf("%lf", &z); // Read a double from the user and store it in z
102 printf("new z=%lf\n", z); // Print the value of z with 2 decimal places
103
104 printf("Enter a char for ch: ");
105 scanf(" %c", &ch); // Read a char from the user and store it in ch
106 // Note the space before %c, it is used to consume the newline character
107 // left in the buffer
108 printf("new ch=%c\n", ch); // Print the value of ch
109
110 // Precision
111 // The precision of a number is the number of digits after the decimal point.
112 // The default precision is 6 digits.
113
114 printf("y=%.2f\n", y); // %.2f is a flag for a float with 2 decimal places
115 printf("z=%.3lf\n", z); // %.3lf is a flag for a double with 3 decimal places
116
117 printf("f=%7.3lf\n", f);
118 // %7.4lf means the total width of the number is 7
119 // including the decimal point and the digits before it
120 // If the number is less than 7 digits, it will be padded with spaces on left
121
122 // left- and right+ alignment in printf
123 printf("f=%-7.3lf\n", f);
124 printf("f=%+7.3lf\n", f);
125 // fill with 0s alignment
126 printf("f=%07.3lf\n", f);
127
128 // More on array and 2d array
129 int NEW[3][4] = {{1, 2, 3, 5}, {4, 6, 7, 9}, {-11, 8, -12, 14}};
130 double NEWD[2][2] = {1.1, 2.2, 3.3, 4.4}; // you can put them just in one {}
131
132 // first [] is row index, second [] is column index
133 printf("the element at row 1 and column 3 of NEW is %d\n", NEW[0][2]);
134
135 // FIXME: printf("row 2 and column 2 of NEWD is %d\n", NEWD[2][2]);
136 printf("the element at row 2 and column 2 of NEWD is %lf\n", NEWD[1][1]);
137
138 // You can also scanf to update a value in the array
139 printf("Enter a new value for NEW[0][2]: ");
140 scanf("%d", &NEW[0][2]);
141 printf("the element at row 1 and column 3 of NEW is %d\n", NEW[0][2]);

```

```
142     printf("\n\n");
143
144     /* 03/13's Content: Escape Sequences
145     =====
146     */
147     printf("03/13's Content: Escape Sequences\n");
148     printf("=====\\n");
149
150     // Using \n to enter (go to new line)
151     printf("Hello, world!\n");
152     printf("This is a new line.\n");
153     printf("\n");
154
155     // How to literally print \n
156     printf("Here is a backslash plus n: \n");
157     printf("\\n");
158
159     // Using \t to insert a tab
160     printf("\n\n");
161     printf("Column1\tColumn2\tColumn3\n");
162     printf("Data1\tData2\tData3\n");
163     printf("\n");
164
165     // How to literally print \t
166     printf("Here is a backslash plus t: \n");
167     printf("\\t");
168
169     // Using \\ to display a backslash
170     printf("\n\n");
171     printf("This is a backslash: \\");
172     printf("\n");
173
174     // Using \' to display a single quote
175     printf("This is a single quote: \' ");
176     printf("\n");
177
178     // Using \" to display a double quote
179     printf("This is a double quote: \" ");
180     printf("\n");
181
182     // Using %% to display a percent sign
183     printf("This is a percent sign: %% ");
184     printf("\n");
185
186     // Using \n\n to show a blank line
187     printf("\nHere is a blank line in between:\n");
188     printf("abcd\nnefgh\n\n\n");
189
```

```

190  /* 03/13's Content: HW7 Hints
191  =====
192  */
193  printf("03/13's Content: HW7 Hints\n");
194  printf("=====\\n");
195
196  // Part 3:
197  // There is no breakline between "allow the printing".
198  // It is the display simply due to pdf rendering
199  // Similarly, this is applied to consequent paragraphs
200  // But there should be between "do you" and "understand" and "the code"
201  printf("do you\\nunderstand\\nthe code");
202
203  /*
204  Part 4:
205  No need to show anything before scans
206  Expected format is:
207
208  <Text>
209  <more text>
210  A=?
211  B=?
212  C=?
213  <more text>
214  A=?
215  B=?
216  C=?
217  <more text>
218  A=?
219  B=?
220  C=?
221
222  Part 5 & 6:
223  Expected format is:
224
225  G=
226  ?
227  ?
228  ?
229  ?
230
231  H=
232  ?
233  ?
234  ?
235  ?
236  */

```

```

237
238 // Can I try the code locally before submission?
239 // Yes, the starter code is using the following structure
240 // which we will discuss in the future classes
241
242 // After compiling and building in Geany, you will prompt for input
243 // Enter 1: you will test your Part 1 code
244 // Enter 2: you will test your Part 2 code
245 printf("\n\n");
246
247 char choice;
248 printf("Demo test, Enter 1 or 2: ");
249 scanf(" %c", &choice); // Read a char from the user and store it in ch
250 switch (choice)
251 {
252     case '1':
253     {
254         printf("You are testing part 1.\n");
255         break;
256     }
257     case '2':
258     {
259         printf("You are testing part 2.\n");
260         break;
261     }
262 }
263 return 0;
264 }
265
266 /*
267 Workflow: Compiling and Running the Program
268
269 1. Write the Program:
270     - Open your text editor or IDE (e.g., Geany, VS Code).
271     - Write the C code as shown below and save it as `week10_tutorial.c`.
272     - If you are using Geany, you can create a c file using template
273     - If you are using the VS Code cloud IDE (i.e., here), compile and run are
just to click the run button
274
275 2. Compile the Program:
276     - Option A : if in Geany, click compile and then click build
277     - Option B : if in computer terminal
278     - Open the terminal, navigate to the directory where your c file stores.
279     - Use the `gcc` compiler to compile the program:
280         gcc -o week10_tutorial week10_tutorial.c
281     - This command tells `gcc` to compile `week10_tutorial.c` and output an
executable named `week10_tutorial`.
282

```

```

283 3. Run the Compiled Program:
284     - Option A : if in Geany, simply click execute
285     - Option B : if in computer terminal
286     - In the terminal, run the compiled program by typing:
287         ./week10_tutorial
288     - This command executes the `week10_tutorial` program, and you should see the
      output in the terminal.
289 */
290
291 /*
292 Output of the program:
293
294 03/11's Content: C Programming Basics, Arrays
295 =====
296 Hello, world!
297 Welcome to learning C programming!
298 The sum of 5 and 10 is 15
299 the fourth element of M is 40
300 the element at row 1 and column 3 of N is 3
301
302
303 03/13's Content: Data Types, Printf, Scanf, Precision
304 =====
305 x=10
306 y=3.140000
307 z=3.141590
308 ch=A
309 option=1
310 z in integer is 1431671456
311 ch in integer is 65
312 Enter an integer for x: 87
313 new x=87
314 Enter a double for z: 7.89
315 new z=7.890000
316 Enter a char for ch: p
317 new ch=p
318 y=3.14
319 z=7.890
320 f= 1.235
321 f=1.235
322 f= +1.235
323 f=001.235
324 the element at row 1 and column 3 of NEW is 3
325 the element at row 2 and column 2 of NEWD is 4.400000
326 Enter a new value for NEW[0][2]: 567
327 the element at row 1 and column 3 of NEW is 567
328
329

```

```
330 03/13's Content: Escape Sequences
331 =====
332 Hello, world!
333 This is a new line.
334
335 Here is a backslash plus n:
336 \n
337
338 Column1 Column2 Column3
339 Data1    Data2    Data3
340
341 Here is a backslash plus t:
342 \t
343
344 This is a backslash: \
345 This is a single quote: '
346 This is a double quote: "
347 This is a percent sign: %
348
349 Here is a blank line in between:
350 abcd
351
352 efgh
353
354
355 03/13's Content: HW7 Hints
356 =====
357 do you
358 understand
359 the code
360
361 Demo test, Enter 1 or 2: 2
362 You are testing part 2.
363
364 */
```