

# 1806ICT Programming Fundamentals

## Workshop Week 2: Data Types, Operators & Expressions

### Part A: PC-based Questions

Please try to complete as many of these exercises before the workshop.

1. Given the following C program:

```
/*
 * This program initializes the value of count, increments it,
 * and prints out the final value of count.
 */
#include <stdio.h>

int main()
{
    Int count;

    /* initialize count
    count = 1
    /* increment count */
    count++;
    print the value of count */
    printf("count = %d\n", count);

    return 0;
}
```

Try to compile the program. If the compilation is unsuccessful, identify the errors in the program, and correct them. Once the program is corrected, compile and run the program again.

2. The following program is given:

```
#include <stdio.h>
int main(
){float qx,
zz,
tt;printf("gimme 3\n"
);scanf
( "%f%f %f", &qx, &zz
, &tt);printf("average is=%f",
(qx+tt+zz)/3.0);return
0
;}
```

Although the code is not very readable, it does compile and run. Test it to see if that is true. Then completely rewrite the program, using white spaces and comments to make it more readable and well documented.

3. C provides the unary operator `sizeof` to find the number of bytes needed to store a variable type. The following program uses this operator. On a given machine, it provides precise information about the storage requirements for the fundamental types. Try the following program on your machine.

```
/*
 * This program computes the size of some fundamental types.
 */
#include <stdio.h>

int main()
{
    printf("char          : %u byte \n", sizeof(char));
    printf("short int     : %u bytes\n", sizeof(short int));
    printf("int             : %u bytes\n", sizeof(int));
    printf("long int        : %u bytes\n", sizeof(long int));
    printf("unsigned int: %u bytes\n", sizeof(unsigned int));
    printf("float          : %u bytes\n", sizeof(float));
    printf("double         : %u bytes\n", sizeof(double));
    printf("long double  : %u bytes\n", sizeof(long double));

    return 0;
}
```

Because the C language is flexible in its storage requirements for the fundamental types, the situation can vary from one machine to another. However, it is guaranteed that

```
sizeof(char) = 1
sizeof(char) ≤ sizeof(short int) ≤ sizeof(int) ≤ sizeof(long int)
sizeof(signed int) = sizeof(unsigned int) = sizeof(int)
sizeof(float) ≤ sizeof(double) ≤ sizeof(long double)
```

4. In C, variables of type `char` are stored as a small integer value. Each char is stored in memory in 1 byte. Most machines represent characters using the ASCII character set, which associates the integers 0 to 127 with specific characters. See [www.asciitable.com](http://www.asciitable.com). Try the following program on your machine:

```
#include <stdio.h>

int main()
{
    char c = 'a';

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

    return 0;
}
```

Now, consider the following program:

```
#include <stdio.h>

int main()
{
    char c = 'A';

    printf("sizeof(c)           = %u\n", sizeof(c));
    printf("sizeof('A')        = %u\n", sizeof('A'));
    printf("sizeof(c + c)       = %u\n", sizeof(c + c));

    return 0;
}
```

What do you think will be printed?

Code the program in CodeLite to check your answers.

Try to explain the output of this program.

5. In the lecture slides, you were introduced to the functions `printf()` and `scanf()`. There is a whole lot more options that are available for formatting data with these functions. Please read the following references:

Kochan, Chapter 15, pages 346 – 358

<https://wpollock.com/CPlus/PrintfRef.htm>

<http://www.cprogramming.com/tutorial/printf-format-strings.html>

6. Try the following code on your machine:

```
#include <stdio.h>

int main()
{
    int bigNumber = 20000000000 + 20000000000;

    printf("%d      %u\n", bigNumber, bigNumber);

    return 0;
}
```

What gets printed? Explain the output. (Note: The output of this program will become a lot more understandable after attending the 1007ICT/1807ICT Week 2 lecture).

7. Given the following program:

```
#include <stdio.h>

int main()
{
    int a, b=0, c=0;

    a = ++b + ++c;
    printf("%d %d %d\n", a, b, c);

    a = b++ + c++;
    printf("%d %d %d\n", a, b, c);

    a = ++b + c++;
    printf("%d %d %d\n", a, b, c);

    a = b-- + --c;
    printf("%d %d %d\n", a, b, c);

    return 0;
}
```

What do you think will be printed?

Code the program in CodeLite to check your answers.

8. Write a program that asks the user to enter two parameters: the `area` of a rectangle (in square m), and the `width` of the rectangle (in m). The program will compute and display the `height` of the rectangle (in m). Assume that the `area`, `width`, and `height` are type double variables.

Sample run:

Input		Output
Area (square m)	Width (m)	Height (m)
5.25	2.5	2.1

9. Write a program that converts an angle in degrees to radians. You can use the following formula:

$$\text{Radians} = (\text{Degrees} * \text{M\_PI}) / 180.0$$

`M_PI` is a mathematical constant for  $\pi$ , defined in the `<math.h>` library, so you will need to `#include <math.h>` in your program.

Sample run:

Input	Output
Degrees	Radians
180.0	3.141593
92.35	1.611812

10. Write a program which accepts a time interval in seconds and prints the equivalent time in hours, minutes, and seconds. One hour is 3600 seconds and one minute is 60 seconds.

Sample Run:

Input	Output		
Seconds	Hours	Minutes	Seconds
3600	1	0	0
5000	1	23	20

11. Given a 24 hour time of day as [hours minutes seconds], add a time interval which is also specified as [hours minutes seconds]. Write a program to compute this addition operation, and print the resultant 24 hour time of day in [hours minutes seconds].

Sample Run:

Input						Output		
24 hour time of day			Time interval			24 hour time of day		
Hours	Mins	Secs	Hours	Mins	Secs	Hours	Mins	Secs
1	0	0	1	30	12	2	30	12
1	15	50	2	15	30	3	31	20
13	24	30	2	40	40	16	5	10

12. The C math library function `sqrt(x)` can be used to determine the square root of  $x$ , where  $x$  is a type `double`. This function is defined in the `<math.h>` library, so you will need to `#include <math.h>` in your program. The following example program shows the usage of the `sqrt()` function.

```
#include <stdio.h>
#include <math.h>

int main ()
{
    printf("Square root of %f is %f\n", 4.0, sqrt(4.0) );
    printf("Square root of %f is %f\n", 5.0, sqrt(5.0) );

    return 0;
}
```

Now, write a program that solves the polynomial equation  $ax^2 + bx + c = 0$

Sample Run:

Input			Output	
a	b	c	X1	X2
1	2	1	-1.0	-1.0
2	4	1	-1.707	-0.293

13. Write a program that computes the score needed on a final exam (out of 100) to achieve a desired grade in a course. The program should interact with the user as follows:

Enter desired grade> B

Enter minimum score required for the desired grade> 79.5

Enter current average score in course> 74.6

Enter how much the final exam counts as a percentage of the course grade> 25

You need a score of 94.20 on the final exam to get a B.

*In the example shown above, the final exam counts as 25% of the course grade.*

Sample Run:

Input				Output
Desired grade	Min score	Current avg. score	Final exam %	
B	79.5	74.6	25	94.2
A	85.0	80.3	50	89.7