Name: _____ Student ID: _____ Date: _____

Worksheet 3

Data Types, Conversion Characters

Objectives

After completing this worksheet, you should be able to

- Know various elements that build C language
- Use an output function (command) in C
- Declare variables with suitable data types
- Understand the conversion specifications in an output function

1. Run the Code::Blocks program, and then create a new project and write Program 3.1.

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
        printf("1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890\n");          ----②
        printf("Hello, everyone.");                                          ----------③
        printf("Welcome to \b\b\b C world.");                                ----------④
        printf("\n\n");                                                      ----------⑤
        printf("\tCollege of Industrial Technology.\n");                     ----------⑥
        printf("KMUT\'NB\n\n");                                              ----------⑦
        printf("print special character \'  \"  \\ ");      // Display ' " and \ marks  ----------⑧
        printf("\a\a\a");                                                    ----------⑨
        return 0;
}
```

Program 3.1

2. Compile and run the program (choose **Execute** ➔ **Compile & Run** or press **F9** key), then record the result produced by the program in Table 3.1.

Line number	Column / Results produced by program 3.1 (1 … 10 … 20 … 30 … 40 … 50)
1	
2	
3	
4	
5	
6	
7	
8	

Table 3.1

3. Modify the escape sequences in the statements ⑤,⑦, ⑧ and ⑨ in Program 3.1 as follows

```
printf("\xa\xa");  ----------⑤
printf("KMUT\x27NB\xa\xa");  ----------⑦
printf("print special character \x27 \x22 \x5c"); ----------⑧
printf("\x7\x7\x7");  ----------⑨
```

Run the modified program. Is the result produced by the program different from one provided by Program 3.1. Why?

_____

_____

_____

4. Open a new editor, write Program 3.2 and then run and record the result.

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int   a = 123;          //Declare an integer variable, named a, and initialize it with 123
    float b = 456.78;       //Declare a floating point variable, named b, and initialize it with 456.78
    char c = 'Z';           // Declare a character variable, named c, and initialize it with Z
    printf("a  =  %d \n",a);  ----------(1)
    printf("b =  %f \n",b);  ----------(2)
    printf("c =  %c \n",c);  ----------(3)
    return 0;
}
```

Program 3.2

4.1) a = ...

4.2) b = ...

4.3) c = ...

5. Change the function (2) in Program 3.2 to

```
printf("b =  %e \n",b);  ----------(2)
```

Run the program and record the result b = _____

What does this function display? _____

_____

6. Declare three variables that can store the vales as follows:
    - The first variable named "**X**" stores the value ranging from -32768 to 32767.
    - The second variable named "**Y**" stores the value ranging from -1.7×10^{-308} to $1.7 \times 10^{+308}$.
    - The third variable named "**Z**" stores one character ranging from a – z or A-Z or +, -, *, / signs.

_____
_____
_____
_____

7. Open a new editor, write Program 3.3 and run it.

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    int x, y;   float z;
    x = 15;  y = 7;  z = 3.141596295;
    printf("12345678901234567890123456789O\n");                    ----------(1)
    printf("%d\n",x);                                              ----------(2)
    printf("%5d\n",x);                                             ----------(3)
    printf("%20d\n",x);                                            ----------(4)
    printf("%010d\n",x);       //This function is equivalent to printf("%.10d\n",x);  ----------(5)
    printf("%*d\n",y,x);   //Add extra spaces corresponding to a value defined in y  ----------(6)
    printf("%f\n",z);  ----------(7)
    printf("%10.3f\n",z);  ----------(8)
    printf("%10.3e\n",z);                                          ----------(9)
    printf("%020f\n",z);  ----------(10)
    printf("%*.*f\n",x,y,z)                ;----------(11)
    system("PAUSE");
    return 0;
}
```

Program 3.3

8. Record the result obtained from Program 3.3 in Table 3.2

Function no.	Column Experimental Results																																							
	1									10										20										30										40
1																																								
2																																								
3																																								
4																																								
5																																								
6																																								
7																																								
8																																								
9																																								
10																																								
11																																								
12																																								

Table 3.2

Name : _____ Student ID _____

Homework 3

1. Which of the following are valid variable names in the C program? If invalid, explain why?

   1.1 root _____

   1.2 main _____

   1.3 Salary _____

   1.4 9thFloor _____

   1.5 input value _____

   1.6 MyAddress _____

   1.7 int _____

   1.8 AUto _____

   1.9 You&me _____

   1.10 income_tax _____

2. Given a set of **integer** variables as follows:

a = 1, b = 100, c = 10000,

x = 9, y = 999, and z = 9999

From the knowledge of the **field width** in C language, write a program that displays the values stored in above variables in table-like format as shown below.

```
-------------------------------------------------------------------
|        Number        |           Amount           |
==========================================
|             1        |              9             |
|           100        |            999             |
|         10000        |           9999             |
==========================================
```