Department of Electrical and Computer Engineering
University of Massachusetts Dartmouth

ECE160: Foundations of Computer Engineering I (Spring 2023)
Instructor: Dr. Liudong Xing

LAB # 2 Solution

1. Type, compile, and run this program, and correct any errors you may find:

```
#include "stdio.h"
 void main(void)
  {
          printf(‘Today is Monday.’)
  }
```

Solution:
printf("Today is Monday.");
Two errors:
1) Single quote should be double quote
2) semicolon ; is missing.

2. C code consists of a number of tokens. A C token is the smallest element that C compiler does not break down into smaller parts. A token can be a function name (e.g., main, printf) or a C keyword (e.g., int, char). All tokens should be written continuously. Between tokens, white space characters (e.g., blank, tab, carriage return) can be inserted, but this is optional. In summary, it is acceptable to add blanks between tokens but not acceptable to add blanks within tokens.

   Type, compile, and run this program, correct any errors you may find.

```
#include "stdio.h"
 void ma  in(void)
 {
          Printf   ("Tomorrow is Saturday."      )   ;     }
```

Solution:
remove the space between ma and in; and change Printf to printf
Two errors:
1) no space within a token
2) Printf is different from printf (printf should be used).

3. Type, compile, and run this program, correct any errors you may find.

```
#include "stdio.h"
int main(void)
{
     float celsius;
     float fahrenheit;
     float temp;
     float constant;
     printf("This program converts Celsius  to Fahrenheit.\n");
     printf("Please enter a Celsius temperature.\n");
     scanf_s("%f", &celsius);
     constant = 9.0/5.0;
     temp = constant *Celsius;
     fahrenheit = Temp+ 32;
     printf("The temperature in Fahrenheit is: %f\n", fahrenheit);
     return 0;
}
```

Solution:

change Celsius to celsius
change Temp to temp

Errors: C is case sensitive. Celsius and celsius are different identifiers; Temp and temp are different identifiers. The one defined at the beginning of the program should be used.

4. Modify the program in problem 3 (after correcting the errors) by changing the Literal Constants 9.0, 5.0, and 32 to **Defined Constants** using names of c1, c2, and c3, respectively. Then compile and run the modified program.

Solution:

```
#include "stdio.h"
#define c1 9.0
#define c2 5.0
#define c3 32

int main(void)
{
        float celsius;
        float fahrenheit;
        float temp;
        float constant;
        printf("This program converts Celsius  to Fahrenheit.\n");
        printf("Please enter a Celsius temperature.\n");
        scanf_s("%f", &celsius);
        constant = c1/c2;
        temp = constant *celsius;
        fahrenheit = temp+ c3;
        printf("The temperature in Fahrenheit is: %f\n", fahrenheit);
        return 0;
}
```

5. Modify the program in problem 3 (after correcting the errors) by changing the Literal Constants 9.0, 5.0, and 32 to **Memory Constants** using the **float** type and names c1, c2, and c3, respectively. Then compile and run the modified program.

Solution:

```
#include "stdio.h"

int main(void)
{
        const float c1 = 9.0;
        const float c2 = 5.0;
        const float c3 = 32;
        float celsius;
        float fahrenheit;
        float temp;
        float constant;
        printf("This program converts Celsius  to Fahrenheit.\n");
        printf("Please enter a Celsius temperature.\n");
        scanf_s("%f", &celsius);
        constant = c1/c2;
        temp = constant *celsius;
        fahrenheit = temp+ c3;
        printf("The temperature in Fahrenheit is: %f\n", fahrenheit);
        return 0;
}
```