```
1   #include<stdio.h>
2   #include<conio.h>
3   #pragma warning(disable:4996)        // Needed to compile when using VS19 scanf()
4   #define TWICE 2
5
6   float dollar_val (int iWeight, float fValue_per_lb);
7   /*--------------------------------------------------------------------*/
8   int main(void)
9   {
10    int iWeight;
11    float fValue_per_lb, fValue, fTwice;
12
13    printf("Enter your value in dollars per pound => ");
14    scanf("%f",&fValue_per_lb);
15
16    printf("\nEnter your weight in pounds => ");
17    scanf("%d",&iWeight);
18
19    fValue = dollar_val (iWeight, fValue_per_lb);
20    fTwice = TWICE * fValue;
21
22    printf("\nYou are worth %f dollars.\n", fValue);
23    printf("You are worth %0.2f dollars. \n", fValue);
24    printf("Twice your worth is %8.2f dollars. \n", fTwice);
25
26    return 0;
27  }
28  /*--------------------------------------------------------------------*/
29  float dollar_val (int iWeight, float fValue_per_lb)
30  {
31    float fAnswer;
32
33    fAnswer = (float)iWeight * fValue_per_lb;
34    return  fAnswer;
35  }
36  /*--------------------------------------------------------------------*/
37  /*
38  Enter your value in dollars per pound => 20
39
40  Enter your weight in pounds => 210
41
42  You are worth 4200.000000 dollars.
43  You are worth 4200.00 dollars.
44  Twice your worth is  8400.00 dollars.
45
46  */
47
```

Commented [KW1]: The #define allows for a constant value to be associated with a label/name. Standard convention is for constants to be in all capital letters.

Commented [KW2]: This is a function prototype for a user generated function. Notice that all prototypes end with a semicolon.

Commented [KW3]: These are actual parameters, which are local variables used in main(). This is called passing by value. Notice when calling the function dollar_val() that the data types must match. Also notice in this particular example the main() and dollar_val() variable names are exactly the same, which is not a requirement nor is it desired.

Commented [KW4]: Replaced by number 2 during the preprocessor action.

Commented [KW5]: In the function definition, the formal parameters are the same as declaring some/all of the function's local variables.

Also notice that the beginning of the function definition must match the prototype exactly with the only exception being that there is no semicolon at the end of the last parenthesis.

Commented [KW6]: This is called casting, which changes the data type/size only for this instance where used.

Commented [KW7]: Lines 13 and 14 output.

Commented [KW8]: Lines 16 and 17 output.

Commented [KW9]: Lines 22 – 24 output.

```
#include<stdio.h>
#include<conio.h>
#pragma warning(disable:4996)        // Needed to compile when using VS19 scanf()
#define TWICE 2

float dollar_val (int iWt, float fVal_per_wt);
/*-------------------------------------------------------------------*/
int main (void)
{
  int iWeight;
  float fValue_per_lb, fValue, fTwice;

  printf("Enter your value in dollars per pound => ");
  scanf("%f",&fValue_per_lb);
  printf("\nEnter your weight in pounds => ");
  scanf("%d",&iWeight);

  fValue = dollar_val (iWeight, fValue_per_lb);
  fTwice = TWICE * fValue;

  printf("\nYou are worth %f dollars.\n", fValue);
  printf("You are worth %0.2f dollars. \n", fValue);
  printf("Twice your worth %8.2f dollars. \n", fTwice);

  return 0;
}
/*-------------------------------------------------------------------*/
float dollar_val (int iWt, float fVal_per_wt)
{
  float fAnswer;

  fAnswer = (float)iWt * fVal_per_wt;
  return  fAnswer;
}
/*-------------------------------------------------------------------*/

/*
Enter your value in dollars per pound => 20

Enter your weight in pounds => 210

You are worth 4200.000000 dollars.
You are worth 4200.00 dollars.
Twice your worth  8400.00 dollars.
*/
```

Commented [KW10]: Everything in this program is exactly the same as the previous program with the exception the formal parameters which have no been changed to be different than the variable names in main(). This is the preferred method when creating variables since it easily allows for the various variables to be identified easily since ALL names are different.

Commented [KW11]: The data returned from dollar_val() is stored locally in main in variable fValue.

```
#include<stdio.h>
#include<conio.h>
#include <stdlib.h>
#pragma warning(disable:4996)        // Needed to compile when using VS19 scanf()

float calc_torque (float fFor, float fMome_Arm);
void display_worth (float fTorq);
/*--------------------------------------------------------------------*/
int main (void)
{
  float fForce, fMoment_Arm, fTorque;

  printf("Enter the force in pounds => ");
  scanf("%f",&fForce);
  printf("\nEnter the moment arm in feet => ");
  scanf("%f",&fMoment_Arm);

  fTorque = calc_torque (fForce, fMoment_Arm);
  display_worth (fTorque);

  return 0;
}
/*--------------------------------------------------------------------*/
float calc_torque (float fFor, float fMome_Arm)
{
  float fAnswer;

  fAnswer = fFor * fMome_Arm;
  return  fAnswer;
}
/*--------------------------------------------------------------------*/
void display_worth (float fTorq)
{
  printf("\nThe Torque is %f foot-pounds.\n", fTorq);
  printf("\nThe Torque is %0.2f foot-pounds.\n", fTorq);
  printf("\nThe Torque is %0.2e foot-pounds.\n", fTorq);
}
/*--------------------------------------------------------------------*/
/*
Enter the force in pounds => 3

Enter the moment arm in feet => 4

The Torque is 12.000000 foot-pounds.

The Torque is 12.00 foot-pounds.

The Torque is 1.20e+01 foot-pounds.
*/

```

Commented [KW12]: Two user generated function prototypes with formal parameters named different than the passed actual parameters.

Commented [KW13]: Various ways to display the same data.

```
#include<stdio.h>
#include<conio.h>
#pragma warning(disable:4996)        // Needed to compile when using VS19 scanf()

#define OFFSET 7.64      /* these are preprocessor directives */
#define SCALE 0.325      /* that define constants */

float get_shoe_size(void);             /* function prototypes */
void display_foot_size(float fValue);
/*-------------------------------------------------------------------*/
int main (void)
{
  float fShoe, fFoot;

  fShoe = get_shoe_size();
  fFoot =  (SCALE * fShoe) + OFFSET;
  display_foot_size(fFoot);

  fShoe = get_shoe_size();
  fFoot =  (SCALE * fShoe) + OFFSET;
  display_foot_size(fFoot);

  return 0;
}
/*-------------------------------------------------------------------*/
float get_shoe_size(void)
{
  float fShoe;

  printf("Enter your Shoe size  => ");
  scanf("%f",&fShoe);
  return(fShoe);
}
/*-------------------------------------------------------------------*/
void display_foot_size(float fValue)
{
  printf("Your foot size is %0.1f inches.\n\n",fValue);
}
/*-------------------------------------------------------------------*/
/*
Enter your Shoe size  => 8
Your foot size is 10.2 inches.

Enter your Shoe size  => 10
Your foot size is 10.9 inches.

*/
```

Commented [KW14]: The formal parameter is void, so no data is passed from main() to this function.

Commented [KW15]: Notice when calling get_shoe_size() that since the formal parameter is void, that the actual parameter field is left blank.

Commented [KW16]: This shows that these functions can be called as many times as needed.

Commented [KW17]: This instance of FShoe is a totally different variable than the FShoe found in main().