

Problem 1

Program	Error Message	Cause of Error	Correction Applied
program01.c	program01.c:17:2: error: unknown type name 'DOUBLE'	Double in all capitals is invalid syntax	Line 17: double x, y, result;
program02.c	program02.c:23:2: warning: 'y' is used uninitialized in this function [-Wuninitialized]	& symbol missing in scanf before the variable name	Line 23: scanf ("%lf", &y);
program03.c	program03.c:6:9: warning: variable 'ave' set but not used [-Wunused-but-set-vari able]	Wrong variable name written in return function.	Line 6: return (ave);
program04.c	program04.c:17:15: warning: variable 'result' set but not used [-Wunused-but-set-vari able]	The aver function shouldn't be printed, the result variable should be printed which runs the aver function with variables x and y	Line 27: printf ("The average is %lf.\n", result);
program05.c	program05.c:25:17: error: incompatible type for argument 1 of 'aver'	The &x is a syntax error because it is not in a scanf function	Line 25: result = aver (x,y);

program06.c	No error message	Arithmetic order of operations error. The parentheses in the function “aver” are required because the two variables must be added first before dividing by 2	Line 8: ave = (n1 + n2) / 2.0;
program07.c	program07.c:8:2: error: expected declaration specifiers before 'ave'	The aver function doesn't have an open curly bracket. The function type should be double because the program should be able to calculate the average for any two numbers, even if one or both of them are rational	Line 3: double Line 5: {
program08.c	program08.c:10:10: error: 'result' undeclared (first use in this function)	There is no such variable called “result” in the aver function. It can not be used as the return value	Line 10: return (ave);
program09.c	program09.c:25:21: error: expected ';' before 'printf'	There is a missing semicolon after the result variable is defined.	Line 25: result = aver (x,y);
program10.c	program10.c:27:2: warning: statement with no effect [-Wunused-value]	Missing opening bracket in the printf statement in the “main” function	Line 27: printf ("The average is %lf.\n", result);
program11.c	program11.c:8:9: error: 'n1' undeclared (first	The variables n1 and n2 are undeclared in	Line 4:

	use in this function)	the aver function.	aver (double n1, double n2)
program12.c	program12.c:20:12: warning: unknown conversion type character '.' in format [-Wformat=]	The 2.2 in front of the %lf place holder, shouldn't be there. It is a syntax error. If the programmer wanted to make the program return an answer with 2 decimal places, they should've written %.2lf. But since it is better for the program to return a more precise average of two numbers, there should be no restriction for the number of decimal places.	Line 20: scanf ("%lf", &x); Line 23: scanf ("%lf", &y);
program13.c	program13.c:25:11: error: too few arguments to function 'aver'	The variable y is missing in the line that defines the "result" variable. The variable y must be included because the aver function requires two inputs for the program to work.	Line 25: result = aver (x,y);

Problem 2:

Algorithm:

1. Include standard io library
2. Include math library
3. Create a double function, which accepts one argument, the temperature.
4. In the function, create variable “fahrenheit”, which converts the inputted celsius value into a value that is compatible with the formula,
5. Create a variable called “in_root”
6. Create variable “a”
7. Create variable “kmh”, that converts the variable “a” to km/h
8. Convert temperature from celsius to fahrenheit by using the following formula and store it in the fahrenheit variable: $Fahrenheit = 1.8t + 32$
9. Set in_root variable equal to the inside of the square root.

$$in_root = ((5*fahrenheit) + 297) / 247$$
10. Set variable a equal to the formula, but substitute in_root variable into formula

$$a = 1086 * \sqrt{in_root}$$
11. The function should return the value in km/h
 - a. $kmh = a * 1.09728$, this is the conversion factor of feet per second to kilometers per hour
12. Create the main function, that takes the input from the user. Create variables “temp”, and “speed”
13. “temp” will be the input variable, use scanf to ask user to input temperature in celsius.
14. Set “speed” to be equal to the return value of the spsound function, which is kmh.
15. Create a print statement that gives the user their answer

Source code:

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

double spsound (double t)
{
    double a, in_root, fahrenheit, kmh;

    fahrenheit = (1.8*t)+32;
    in_root = ((5*fahrenheit)+297)/247;
    a = 1088*sqrt(in_root);

    kmh = a*(1.09728);
    return (kmh);
}

int main(void)
{
    double temp, speed;

    printf("Enter the temperature in celsius \n");
    scanf("%lf", &temp);

    speed = spsound(temp);

    printf("the speed of sound in %lf degrees celsius is %lf
kilometers per hour", temp, speed);

    return(0);
}
```

Screenshot of code:

```

1  #include <stdio.h>
2  #include <math.h>
3
4
5  double spsound (double t)
6  {
7      double a, in_root, fahrenheit, kmh;
8
9      fahrenheit = (1.8*t)+32;
10     in_root = ((5*fahrenheit)+297)/247;
11     a = 1086*sqrt(in_root);
12
13     kmh = a*(1.09728);
14     return (kmh);
15 }
16
17 int main(void)
18 {
19     double temp, speed;
20
21     printf("Enter the temperature in celsius \n");
22     scanf("%lf", &temp);
23
24     speed = spsound(temp);
25
26     printf("the speed of sound in %lf degrees celsius is %lf kilometers per hour", temp, speed);
27
28     return(0);
29 }
30

```

Screenshot of code execution:

```

C:\Windows\SYSTEM32\cmd.exe
Enter the temperature in celsius
47
the speed of sound in 47.000000 degrees celsius is 2249.262528 kilometers per hour

-----
(program exited with code: 0)
Press any key to continue . . .

```

Proof that it is the correct answer:

47 celsius is 116.6 fahrenheit

$$a = 1086 \sqrt{\frac{(5 \cdot 116.6) + 297}{247}}$$

$$a = 2049.85 \text{ feet/second}$$

$$a = (2049.85)(1.09728)$$

$$a = 2249.259408 \text{ km/h}$$