Function Examples

Lecture 19

Robb T. Koether

Hampden-Sydney College

Wed, Oct 10, 2018

1. Convert Fahrenheit to Celsius
   - The Problem
   - The Input and Output
   - An Example
   - The Algorithm
   - The Code

2. Find the Dimensions of a Rectangle
   - The Problem
   - The Input and Output
   - An Example
   - The Algorithm
   - The Code

Outline

Outline

The Problem

The Problem

Write a program that will compute the equivalent Celsius temperature of each Fahrenheit temperature within a specified range. The program will print the Fahrenheit in that range and its Celsius equivalent, with the Celsius temperature rounded to the nearest 10th of a degree.

Outline

The Input and Output

- Read two temperatures: the starting temperature and the ending temperature, in Fahrenheit.
- Output each Fahrenheit temperature in the range, print it and its Celsius equivalent, round to the nearest 0.1.

Outline

Example

- Let the starting temperature be 65° F and the ending temperature 70° F.

Example

- Let the starting temperature be 65° F and the ending temperature 70° F.
- We must compute the Celsius equivalent of 65, 66, 67, 68, 69, and 70 F.

Example

- Let the starting temperature be 65° F and the ending temperature 70° F.
- We must compute the Celsius equivalent of 65, 66, 67, 68, 69, and 70 F.
- For 65 F, the calculation is

$$\begin{aligned} \text{temp in C} &= (65 - 32) \cdot 5/9 \\ &= 33 \cdot 5/9 \\ &= 165/9 \\ &= 18.3333\ldots \end{aligned}$$

Example

- Let the starting temperature be 65° F and the ending temperature 70° F.
- We must compute the Celsius equivalent of 65, 66, 67, 68, 69, and 70 F.
- For 65 F, the calculation is

$$\begin{aligned} \text{temp in C} &= (65 - 32) \cdot 5/9 \\ &= 33 \cdot 5/9 \\ &= 165/9 \\ &= 18.3333\ldots \end{aligned}$$

- The output should be 65 and 18.3.

Example

- Let the starting temperature be 65° F and the ending temperature 70° F.
- We must compute the Celsius equivalent of 65, 66, 67, 68, 69, and 70 F.
- For 65 F, the calculation is

$$\begin{aligned} \text{temp in C} &= (65 - 32) \cdot 5/9 \\ &= 33 \cdot 5/9 \\ &= 165/9 \\ &= 18.3333\ldots \end{aligned}$$

- The output should be 65 and 18.3.
- Do the same for the other temperatures.

Outline

The Algorithm

- Begin with the starting temperature. Call it "*f_temp*."

The Algorithm

- Begin with the starting temperature. Call it "*f_temp*."
- Subtract 32 from *f_temp* and multiply by 5/9.

The Algorithm

- Begin with the starting temperature. Call it "*f_temp*."
- Subtract 32 from *f_temp* and multiply by 5/9.
- Round off the result to one decimal place. Call it *c_temp*.

The Algorithm

- Begin with the starting temperature. Call it "*f_temp*."
- Subtract 32 from *f_temp* and multiply by 5/9.
- Round off the result to one decimal place. Call it *c_temp*.
- Output *f_temp* and *c_temp*.

The Algorithm

- Begin with the starting temperature. Call it "*f_temp*."
- Subtract 32 from *f_temp* and multiply by 5/9.
- Round off the result to one decimal place. Call it *c_temp*.
- Output *f_temp* and *c_temp*.
- Add 1 to *f_temp*.

The Algorithm

- Begin with the starting temperature. Call it "*f_temp*."
- Subtract 32 from *f_temp* and multiply by 5/9.
- Round off the result to one decimal place. Call it *c_temp*.
- Output *f_temp* and *c_temp*.
- Add 1 to *f_temp*.
- Repeat the previous three steps until *temp* exceeds the ending temperature.

Outline

The Code

- We will use two functions.
  - `float FtoC(int f_temp);`
  - `float round(float value, int places);`
- The function `FtoC()` will convert the Fahrenheit temperature and return its Celsius equivalent, rounded to one decimal place.
- The function `round()` will round the first parameter to the number of decimal places specified by the second parameter.

The Code

- Clearly, we should use a `for` loop to step through the Fahrenheit temperatures.
- For each temperature, we will call on the `FtoC()` function to get the Celsius equivalent.
- Then output the two temperatures.

The Code

The Code – `main()`

```
int main()
{
    int start_temp;
    int end_temp;
    for (int f_temp = start_temp; f_temp <= end_temp; f_tem
    {
        float c_temp = FtoC(f_temp);
        cout << f_temp << "   " << c_temp << endl;
    }
    return 0;
}
```

The Code

The Code – `FtoC()`

```
float FtoC(int f_temp)
{
    float c_temp = (f_temp - 32)*5.0/9.0;
    return round(c_temp, 1);
}
```

The Code

The Code – `round()`

```
float round(float value, int places)
{
    float shift = power(10.0, places);
    float temp = value*shift;
    temp = int(value)/shift;
    return temp;
}
```

Outline

Outline

The Problem

The Problem

- Given the area and perimeter of a rectangle, find the length and width of that rectangle.

Outline

The Input and Output

The Problem

- The input is the area and perimeter of the rectangle.
- The output is the length and width of the rectangle.

Outline

Example

Example

- Let the area be 24 and the perimeter be 20.

Example

Example

- Let the area be 24 and the perimeter be 20.
- Let the length be x and the width be y.

Example

Example

- Let the area be 24 and the perimeter be 20.
- Let the length be x and the width be y.
- Then $xy = 24$ and $x + y = 10$ (half the perimeter).

Example

Example

- Let the area be 24 and the perimeter be 20.
- Let the length be x and the width be y.
- Then $xy = 24$ and $x + y = 10$ (half the perimeter).
- How do we find x and y?

Example

Example

- Solve the equations for x by eliminating y.

Example

Example

- Solve the equations for x by eliminating y.
- We have $y = 10 - x$, so

$$\begin{aligned} 24 &= x(10 - x) \\ 24 &= 10x - x^2 \\ x^2 - 10x + 24 &= 0. \end{aligned}$$

Example

Example

- Solve the equations for x by eliminating y.
- We have $y = 10 - x$, so

$$\begin{aligned} 24 &= x(10 - x) \\ 24 &= 10x - x^2 \\ x^2 - 10x + 24 &= 0. \end{aligned}$$

- We can factor that equation as

$$(x - 4)(x - 6) = 0.$$

Example

Example

- Solve the equations for x by eliminating y.
- We have $y = 10 - x$, so

$$\begin{aligned} 24 &= x(10 - x) \\ 24 &= 10x - x^2 \\ x^2 - 10x + 24 &= 0. \end{aligned}$$

- We can factor that equation as

$$(x - 4)(x - 6) = 0.$$

- Thus, $x = 4$ or $x = 6$.

Example

Example

- Solve the equations for x by eliminating y.
- We have $y = 10 - x$, so

$$\begin{aligned} 24 &= x(10 - x) \\ 24 &= 10x - x^2 \\ x^2 - 10x + 24 &= 0. \end{aligned}$$

- We can factor that equation as

$$(x - 4)(x - 6) = 0.$$

- Thus, $x = 4$ or $x = 6$.
- And so $y = 6$ or $y = 4$.

Example

Example

- In the example, we solved the quadratic equation

$$L^2 - 10L + 24 = 0$$

by factoring it.

- To solve it in a program, we should use the quadratic formula:

$$\begin{aligned} L &= \frac{10 \pm \sqrt{10^2 - 4(1)(24)}}{2(1)} \\ &= \frac{10 \pm \sqrt{4}}{2} \\ &= 5 \pm 1 \\ &= 4 \text{ or } 6. \end{aligned}$$

Outline

The Algorithm

The Algorithm

- Let b be half the perimeter and c be the area.

The Algorithm

The Algorithm

- Let b be half the perimeter and c be the area.
- Solve the quadratic equation

$$x^2 - bx + c = 0.$$

The Algorithm

The Algorithm

- Let b be half the perimeter and c be the area.
- Solve the quadratic equation

$$x^2 - bx + c = 0.$$

- The solutions are

$$L = \frac{-b \pm \sqrt{b^2 - 4c}}{2}.$$

The Algorithm

The Algorithm

- Let b be half the perimeter and c be the area.
- Solve the quadratic equation

$$x^2 - bx + c = 0.$$

- The solutions are

$$L = \frac{-b \pm \sqrt{b^2 - 4c}}{2}.$$

- Find $y = \text{area}/x$.

Outline

The Code

The Code

- We will break the program up into functions.
  - `int main()`
  - `void findLenWid(float area, float perim, float& len, float& wid)`
  - `void solveQuadEq(float a, float b, float c, float& root1, float& root2)`

The Code

The Code – `main()`

```
int main()
{
    float area;
    float perim;
    cin >> area >> perim;
    float length;
    float width;
    findLenWid(area, perim, length, width);
    cout << "Length = " << length;
    cout << "width = " << width << endl;
    return 0;
}
```

The Code

The Code – `findLenWid()`

```
void findLenWid(float area, float perim,
float& len, float& wid)
{
    solveQuadEq(1.0, -perim/2.0, area, len, wid);
    return;
}
```

The Code

The Code – `solveQuadEq()`

```
void solveQuadEq(float a, float b, float c,
float& root1, float& root2)
{
    float temp = sqrt(b*b - 4.0*a*c);
    root1 = (-b + temp)/(2.0*a);
    root2 = (-b - temp)/(2.0*a);
    return;
}
```