

# Problem Solving & Program Design in C

## Chapter 3: Top-Down Design with Functions

### Parts of a Program

- Preprocessor statements
  - Include external libraries
  - Define symbolic constants
  - Define macros
- Function prototypes
- Main function
- Other functions

## Program Development

- Understand Problem
- Analysis
- Data requirements
  - Constants, Input, output and Formulas
- Design
  - Initial algorithm
  - Refine algorithm
- Implementation (Coding)
- Testing
- Maintenance

## C Functions

- `int main(int argc, char* argv[]){}`
- `printf();`
- `scanf();`
- `pow();`

## C Function Libraries

- `#include <header.h>`
- `stdio.h`
- `math.h`
- `string.h`
- `stdlib.h`
- `ctype.h`

## Design Considerations

- Top-Down
  - Consider the execution of `main()`
  - Map function calls
  - Hierarchal diagram
  - Stub functions for testing
- Bottom-Up
  - Consider functions first
  - Analysis and synthesis
  - Code and test functions as units

## Using Functions

- Advantages
  - Procedural abstraction
  - Reuse of code
- Prototypes
- No return and no parameters
- No return with parameters (single & mult)
- Return with parameters

## Documentation

- Paragraph describing function
- Pre-condition and post-condition
- Meaningful variable names
- Comments
- Indentation

## Design a Program

- Determine data needed and develop algorithm

Convert kilometers to miles

Data needed

Floating-point variable for miles

Floating-point variable for kilometers

Algorithm

Get the distance in miles

Convert the distance to kilometers

Kilometers = 1.609 \* Miles

Display the distance in kilometers

## Code the Program

```
// Convert kilometers to miles
#include <stdio.h> // Include standard input/output
#define KMS_PER_MILE 1.609 // Define conversion constant

// Main function
int main(){

    // Data needed
    // Floating-point variable for miles
    double miles = 0.0;
    // Floating-point variable for kilometers
    double kms = 0.0;

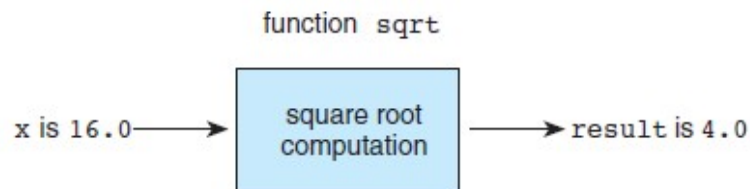
    // Get the distance in miles
    printf("Please enter distance in miles: ");
    scanf("%lf", &miles);

    // Convert the distance to kilometers
    // Kilometers = 1.609 * Miles
    kms = KMS_PER_MILE * miles;

    // Display the distance in kilometers
    printf("%.3f miles is %.3f kilometers.\n", miles, kms);

    return 0;
}
```

## Function sqrt as a “Black Box”



## Modular Solution

- Use a function

```

// Convert kilometers to miles
// Include standard input/output
#include <stdio.h>
// Define conversion constant
#define KMS_PER_MILE 1.609

// Function prototype
void miles_to_kms();

// Main function
int main(){

    miles_to_kms();
    miles_to_kms();
    miles_to_kms();

    return 0;
}

```

```

// Function to get, calculate and print miles to
void miles_to_kms(){
    // Data needed
    // Floating-point variable for miles
    double miles = 0.0;
    // Floating-point variable for kilometers
    double kms = 0.0;

    // Get the distance in miles
    printf("Please enter distance in miles: ");
    scanf("%lf", &miles);

    // Convert the distance to kilometers
    // Kilometers = 1.609 * Miles
    kms = KMS_PER_MILE * miles;

    // Display the distance in kilometers
    printf("%.3f miles is %.3f kilometers.\n", miles, kms);
}

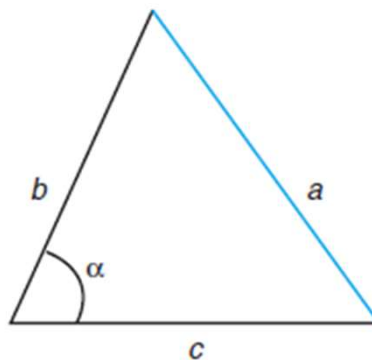
```

## Some Simple Functions

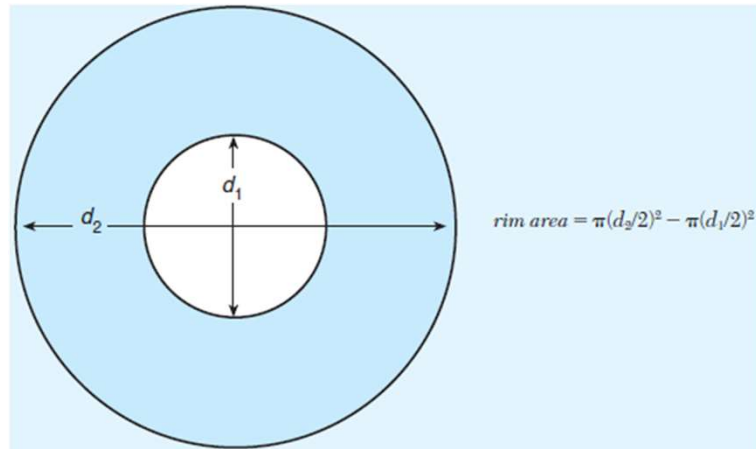
- Calculating the area and circumference of a circle
  - $\text{area} = \pi r^2$
  - $\text{circumference} = 2\pi r$
- Calculate the hypotenuse of a triangle
  - $a^2 = b^2 + c^2$
- Calculate the distance an object falls in a vacuum
  - $y = 1/2 G t^2$

## Compute Unknown Side of Triangle

- $a^2 = b^2 + c^2 - 2bc \cos \alpha$
- Trig functions in C require radians



## Computing the Rim Area of a Flat Washer



## Calculate the Shipping Weight of a Case of Washers

- Calculate area of a circle
- Calculate rim area of a washer
- Calculate volume of a washer
- Calculate mass of a washer
- Calculate the mass of a case of washers