

Finding the Area and Circumference of a Circle

```
1. /*
2.  * Calculates and displays the area and circumference of a circle
3.  */
4.
5. #include <stdio.h> /* printf, scanf definitions */
6. #define PI 3.14159
7.
8. int
9. main(void)
10. {
11.     double radius;    /* input - radius of a circle */
12.     double area;     /* output - area of a circle */
13.     double circum;   /* output - circumference */
14.
15.     /* Get the circle radius */
16.
17.     /* Calculate the area */
18.     /* Assign PI * radius * radius to area. */
19.
20.     /* Calculate the circumference */
21.     /* Assign 2 * PI * radius to circum */
22.
23.     /* Display the area and circumference */
24.
25.     return (0);
26. }
```

Program Design

```

1.  /*
2.   * Calculates and displays the area and circumference of a circle
3.   */
4.
5.  #include <stdio.h> /* printf, scanf definitions */
6.  #define PI 3.14159
7.
8.  int
9.  main(void)
10. {
11.     double radius; /* input - radius of a circle */
12.     double area;   /* output - area of a circle   */
13.     double circum; /* output - circumference     */
14.
15.     /* Get the circle radius */
16.     printf("Enter radius> ");
17.     scanf("%lf", &radius);
18.
19.     /* Calculate the area */
20.     area = PI * radius * radius;
21.
22.     /* Calculate the circumference */
23.     circum = 2 * PI * radius;
24.
25.     /* Display the area and circumference */
26.     printf("The area is %.4f\n", area);
27.     printf("The circumference is %.4f\n", circum);
28.
29.     return (0);
30. }

```

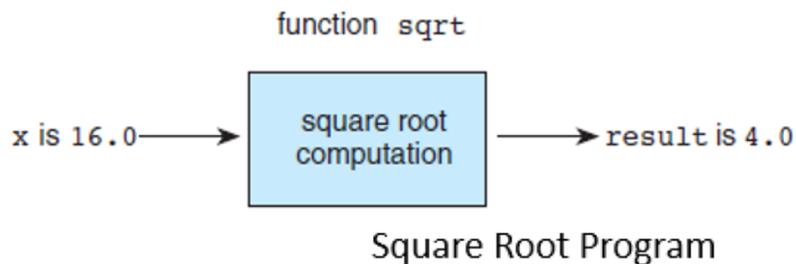
Enter radius> 5.0
The area is 78.5397
The circumference is 31.4159

Program Implementation

Library Functions

- code reuse
 - reusing program fragments that have already been written and tested
- C standard libraries
 - many predefined functions can be found here

Function `sqrt` as a “Black Box”



```
1. /*
2.  * Performs three square root computations
3.  */
4.
5. #include <stdio.h> /* definitions of printf, scanf */
6. #include <math.h> /* definition of sqrt          */
7.
8. int
9. main(void)
10. {
11.     double first, second, /* input - two data values      */
12.           first_sqrt,    /* output - square root of first */
13.           second_sqrt,   /* output - square root of second */
14.           sum_sqrt;      /* output - square root of sum   */
15.
16.     /* Get first number and display its square root. */
17.     printf("Enter the first number> ");
18.     scanf("%lf", &first);
19.     first_sqrt = sqrt(first);
20.     printf("The square root of the first number is %.2f\n", first_sqrt);
```

Note: To enter two variables:

```
#include <math.h>
#include <stdio.h>
void main()
{
    double x, y;
    printf("Enter two doubles...");
    scanf_s("%lf %lf", &x, &y);
    printf("x = %lf, y = %lf", x, y);
}
```

Intro to Functions

Top down design aka stepwise refinement aka divide and conquer

One way to do design is to break a complex problem into simpler problems. For example, a word processor may have a text editor, spelling checker, grammar checker, allow the user a variety of fonts, provide template styles, etc. Each of these smaller tasks can be further broken down into still smaller tasks until we get to a list of tasks that are no larger than a typical homework assignment in this class.

C provides a structure called a *function* to make this possible. Functions are activated by using their name in a program. Most functions have one or more arguments and return a value.

C has a number of built in functions that are provided in a library. For example the math functions are in the math library.

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

double x, y;
x = 22.3;
y = sqrt(x);
```

sqrt is the name of the built-in square root function. x is the argument. The square root function returns a value which is stored in y.

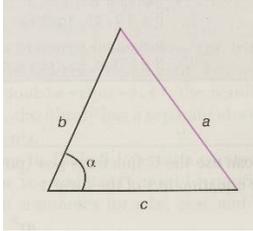
You can have multiple arguments in some functions and they must be separated by commas. For example the pow function takes two arguments.

```
y = pow(x, 3); //y = x^3
```

P. 121 has a short list of some of the more often used math functions.

Note that the trigonometric functions work with radians – not degrees.

Example: Given the length of two sides of a triangle plus the angle between them in degrees, find the length of the third side.



$$a^2 = b^2 + c^2 - 2bc \cos(\alpha)$$

```
a = sqrt(pow(b,2) + pow(c,2) - 2*b*c*cos(alpha*PI/180.0));
```

The functions sqrt, cos, and pow are in math.h

TABLE 3.1 Some Mathematical Library Functions

Function	Standard Header File	Purpose: Example	Argument(s)	Result
abs(x)	<stdlib.h>	Returns the absolute value of its integer argument: if x is -5, abs(x) is 5	int	int
ceil(x)	<math.h>	Returns the smallest integral value that is not less than x: if x is 45.23, ceil(x) is 46.0	double	double
cos(x)	<math.h>	Returns the cosine of angle x: if x is 0.0, cos(x) is 1.0	double (radians)	double
exp(x)	<math.h>	Returns e ^x where e = 2.71828...: if x is 1.0, exp(x) is 2.71828	double	double
fabs(x)	<math.h>	Returns the absolute value of its type double argument: if x is -8.432, fabs(x) is 8.432	double	double
floor(x)	<math.h>	Returns the largest integral value that is not greater than x: if x is 45.23, floor(x) is 45.0	double	double
log(x)	<math.h>	Returns the natural logarithm of x for x > 0.0: if x is 2.71828, log(x) is 1.0	double	double
log10(x)	<math.h>	Returns the base-10 logarithm of x for x > 0.0: if x is 100.0, log10(x) is 2.0	double	double
pow(x, y)	<math.h>	Returns x ^y . If x is negative, y must be integral: if x is 0.16 and y is 0.5, pow(x, y) is 0.4	double, double	double
sin(x)	<math.h>	Returns the sine of angle x: if x is 1.5708, sin(x) is 1.0	double (radians)	double
sqrt(x)	<math.h>	Returns the nonnegative square root of x (√x) for x ≥ 0.0: if x is 2.25, sqrt(x) is 1.5	double	double
tan(x)	<math.h>	Returns the tangent of angle x: if x is 0.0, tan(x) is 0.0	double (radians)	double

P. 121 of the text

Example Problem

Write a program to input a double named x from the user. We will assume that the value entered will be between 0 and 2π . Calculate and print the value of y where y is given by:

$$y = 3x^3 - 4 \cos(2x) + |x - 2| \sin(3x)$$