CSE 105, Lab 3, Summer 2006
cse.unl.edu/∼cstrope/csce105su06/lab/lab3/

Instructor: Cory Strope

July 18, 2006

1. Reusing code
   - Less effort, (Why re-invent the wheel!)
   - Less chance of errors (using previously debugged code, assuming it was debugged)
2. Math library
   - Header file to include as preprocessor directive: <math.h>
   - List of functions available from this library is given on page 109 of text book
3. User defined function - general outline:

```
return_type function_name( arg_type arg_1, arg_type arg_2, ...)
{
    Declare variables used inside the function that are not arguments (multiple lines )
    Executable statements ( could call other functions ) multiple lines
    return value (if function has a return value)
}
```

4. User defined function - itemized explanations:
   - Return type: this is the type of the value the function returns (`int, double, char.` etc.) If the function does not have a returned value, the return type is `void`.
   - Function name: the name by which the function is called (invoked, used). No spaces are allowed in the function name.

- Arguments: List the argument type and name of all values the function needs to perform its task. The type must precede each name separated by a space. If more than one argument is needed, the type-name pair of each value must be separated by a comma. These arguments are enclosed in parenthesis.
- The body of the function starts with an opening brace, {, and ends with the closing brace, }. The local variable declarations, executable statements and `return` statement are written between these braces.
- Local variables (discussed later) need to be declared here. For the time being, we will say that the function can see (use, or change the value of) only the variables in the argument list of the function. So, any other variable to be used must be declared before it is used.
- After the local variables are declared, the task of the function (math calculations, data sorting, etc.) can be performed using the appropriate statements.
- If the function has a return value, the last line in the function must contain the `return` statement followed by an expression or variable that has the same type named in the first line of the function definition.

5. Types of user defined functions:

   - No argument, no return value

     ```
     void function name( void )
     {
         Executable statements (generally to print a standard message to the screen )
     }
     ```

     Compile the source code `ch3sam1.c` and run to see what it does.
     Open source code to examine function structure and compare with outline.

   - Single argument, no return value

     ```
     void function name( argument type argument name )
     {
         Declare needed variables.
         Executable statements (examples applications: file access, calculation that
         are reported to the screen but dont need to be saved, etc.)
     }
     ```

     Compile the source code `ch3sam2.c` and run to see what it does.
     Open source code to examine function structure and compare with outline.

- Multiple arguments, no return value

  ```
  void function name( argument type argument name, argument type argument name)
  {
      Declare needed variables.
      Executable statements (examples applications: file access, calculation that
      are reported to the screen but dont need to be saved, etc.)
  }
  ```

  Compile the source code `ch3sam3.c` and run to see what it does.
  Open source code to examine function structure and compare with outline.

- Multiple arguments, single return value
  See general outline

  Compile the source code `ch3sam4.c` and run to see what it does.
  Open source code to examine function structure and compare with outline.

6. Function prototypes:

   - This is the declaration of the user defined functions you have defined within the source code file.
   - This comes immediately following the preprocessor directives, before the main function.
   - Consists of the first line in the function definition followed by a semicolon!

7. Calling a function:

   - The function is called by the function name followed by a list of variables that contain the values that match the argument list of the function in **n**umber, **o**rder and **t**ype, (not).
   - This comes immediately following the preprocessor directives, before the `main` function.
   - Consists of the first line in the function definition followed by a semicolon!
   - If the function returns no value (type `void`), then no assignment is needed in conjunction with the function call (see `ch3sam1.c`).
   - However, if the function returns a value, and the value is needed in calculations further down the line in the program, than the return value must be stored in a variable of the same type as the return type of the function (see `ch3sam4.c`).

8. Programming problems:

(a) You have saved $500 to use as a down payment on a car. Before beginning your car shopping, you decide to write a program to help you figure out what your monthly payment will be, given the car's purchase price, the monthly interest rate, and the time period over which you will pay back the loan. The formula for calculating your payment is

$$\text{payment} = \frac{iP}{1-(1+i)^{-n}} \, ,$$

where P = principal (the amount you borrow), i = monthly interest rate (1/12 of the annual rate), and n = the number of payments.

Your program should prompt the user for the purchase price, the down payments (usually 36, 48, or 60). It should then display the amount borrowed and the monthly payments including a dollar sign and two decimal places.

(b) The ratio between successive speeds of a six-speed gearbox is

$$\sqrt[3]{M/m} \, ,$$

where M is the maximum speed in revolutions per minute and m is the minimum speed. Write a function `speeds_ratio` that calculates this ratio for any maximum and minimum speeds. Write a main function that prompts for maximum and minimum speeds (rpm), calls `speeds_ratio` to calculate the ratio, and displays the results in a sentence of the form:

```
The ratio between successive speeds of a six-speed gearbox with
maximum speed ________ rpm and minimum speed ________ rpm is ________.
```

Email programming problems a. and b. by midnight tonight!!