

## Day 5 Packaging Code in Functions

**function-** named independent section of code that performs a specific task and optionally returns a value to the calling program

**function prototype-** name, list of variables that CAN be passed to it, and type of variable it returns provides compiler with description of a function that will be defined at a later point. always end with a semicolon

return-type function\_name( arg-type name-1, argtype2 name-2 );

**function definition-** the function itself CAN be placed below main- contains header, statements, local variables (if you want), and return statement. header should be identical to prototype

return-type function\_name( arg-type type-name1 ... ) { statements; }

**argument-** program data sent to the function

if return type anything other than void, needs a return statement

return can be local variable or set of actions. puts can be a return

**PLAN** when writing programs- top down approach

for each argument that is passed to a function **parameter** list must contain one entry

**local variables-** private to that function

function parameters are considered variable definitions so variables in parameter list are also available

local variables can have same names as global variables, but local variables will take precedence

when execution reaches return statement expression is evaluated and result passed back to calling line

return statement terminates the function. can have multiple return statements however without some control first one is executed

stdio.h header is for standard input/output functions

function called with name and arguments- name(arg1)

once declared, function can be used where any C expression used

**recursion-** function calls itself either directly or indirectly

**indirect recursion-** one function calls another function then calls the first function

C allows recursion

inline functuion: inline int function(arg1). inline declaration good for short functions trying to make code more efficient- compiler will try to move function block to call