CSE 115L: Programming Language I Lab (Section: 06)

Spring 2020

Lab Week-04-Function

Function	**Calling Function from main()**

return_type function_name (**parameter_list**)
{
    local variable declaration;
    executable statement1 ;
    executable statement2 ;
    . . . . . .
    return statement;
}
	return_type function_name (**parameter_list**);
	
	global variable declaration (if any);
	
	int main(void) // or any other function
	{
	    function call (**argument list**);
	}

return-type: int, float, double, char etc.
parameters: type name1, type name2,.... Where type can be int, float , char , double and all the basic data types in C.
function declaration or prototype: A function declaration tells the compiler about a function name and how to call the function.
    return_type function_name (parameter list);
** Parameter list must contain their type, may or may not include the variable names

Function with return value	**Function with no return value (void)**

#include <stdio.h>
 	
float average(int first, int second);
// prototype
int main(void)
{
    int a = 7, b = 8;
    float avg = average(a,b);
    printf("%f\n", avg);
    // printing the value which is returned by function
    printf("%f\n", average(a,b));
    // printing the value which is returned by function
    return 0;
}
float average(int first, int second)
{
    return (first+second)/2.0;}
	#include <stdio.h>
	
	void average(int first, int second);
	
	int main(void)
	{
	    int a = 7, b = 8;
	    average(a,b); // calling the function
	    return 0;
	}
	
	void average(int first, int second)
	{
	    printf("%f", (first+second)/2.0);
	}

Some Useful C Library Functions:

Function	Header	Purpose	Argument(s)	Result
abs(x)	<stdlib.h>	Returns the absolute value of its integer arguments	int	int
ceil(x)	<math.h>	Returns the smallest integral value that is not less than x	double	double
pow(x,y)	<math.h>	Returns x raised to the power of y	double	double
cos(x)	<math.h>	Returns the cosine of angle x	double (radian)	double
sqrt(x)	<math.h>	Returns the non negative square root of x for x>= 0.0	double	double

Example: write a function called **checkEvenOdd(int n)** that will determine whether a number is even or odd and print the result with an appropriate message.

```
#include <stdio.h>

void checkEvenOdd(int n);

int main(void)
{
        int number;
        printf("Enter a number: ");
        scanf("%d", &number);
        checkEvenOdd(number);
        return 0;
}

void checkEvenOdd(int n)
{
        //code for determining even or odd
}
```

Example: Write a function that checks whether a particular year is a leap year or not. To determine whether a year is a leap year or not use the following rule.

A leap year must satisfy any or both of the following conditions:

- Divisible by 400
- Divisible by 4 and not divisible by 100

```
#include <stdio.h>

void isLeapYear(int y);

int main(void)
{
    int year;
    printf("Enter a year: ");
    scanf("%d", &year);
    isLeapYear(year);
    return 0;
}

void isLeapYear(int y)
{
    //code for checking if it is a leap year
}
```

Example: Write a function that returns the maximum of the three integers. Take numbers as input.

```
#include <stdio.h>

int maximum(int a, int b, int c);

int main(void)
{
    int a, b, c, max;
    printf("enter three integers: ");
    scanf("%d %d %d", &a, &b, &c);

    max = maximum(a, b, c);
    printf("Greatest number between %d, %d and %d: %d", a, b, c, max);

    return 0;
}
```

```
int maximum(int a, int b, int c)
{
    //code for determining the max number
}
```