Building a Well Structured Program

Using functions and managing memory

Research and Innovation Support, IT-Services

Learning Outcomes

- Using and Writing Functions
- Memory management
  - Introduction to pointers
- Function calls
- Arrays and Structures
- Dynamic memory allocation
- Understanding function calls - the function pointer

Functions

- Group - functions enable grouping of commonly used code into a reusable and compact unit.
- Modularise - programs containing many functions main should be implemented as a group of calls to functions undertaking the bulk of the work
- Reuse - become familiar with rich collections of functions in the ANSI C standard library
- Portability - using functions from ANSI standard library increases portability

```
/*Use the printf function to Display a welcome message on the users screen*/
printf("Welcome to the C Language!\n);"
```

```
printf("b=%f fb=%f\n",b,fb);
```

Overview the input output functions

- Printf
- Scanf
- Format specifiers
- The address operator used in scanf

printf

- Provides formatted input and output
- Input for printf is a format specification followed by a list of variable names to be displayed
- Note the use of escape characters e.g. \n generates a newline

```
printf(“variable %d is %f\n”, myint, myfloat);
```

Examples

```
/*Use the printf function to Display a welcome message on the users screen*/
printf("Welcome to the C Language!\n);"


printf("b=%f fb=%f\n",b,fb);
```

scanf

- Provided an input format and a list of variables
- scanf(“%d”, &myint);
- Note variable name has & in front

```
/*Request input from the user*/
printf("Enter the first integer\n");
scanf("%d", &i1);       /*Read in the integer*/
printf("Enter the second integer\n");
scanf("%d", &i2);
```

Escape Characters

Escape Sequence	**Description**
\n	Newline, position cursor at the start of a new line
\t	Horizontal tab, move cursor to the next tab stop
\r	Carriage return. Position cursor to the beginning of the current line; do not advance to the next line.
\a	Alert, sound system warning beep
\\	Backslash, print a backslash character in a printf statement
\”	Double quote print a double quote character in a printf statement.

Format Specifiers for printf and scanf

Data Type	**Printf specifier**	**Scanf specifier**
long double	%Lf	%Lf
double	%f	%lf
float	%f	%f
unsigned long int	%lu	%lu
long int	%ld	%ld
unsigned int	%u	%u
int	%d	%d
short	%hd	%hd
char	%c	%c

Practice Session

```
int main(char **argv, int argc)
{

        float x,fx;
        float a = 0;
        float fa = -FLT_MIN;
        float b = 3;
        float fb = FLT_MAX;
```

```
  while( fabs(b-a)>(FLT_EPSILON*b))
  {
    x = (a+b)/2;
    /*The function whose root is to be determined*/
    fx = pow(x,3)-2*x-5;
    if(sign(fx)==sign(fa))
    {
      a = x;
      fa = fx;
printf("a=%f fa=%f\n",a,fa);
    }
    else
    {
      b = x;
      fb = fx;
printf("b=%f fb=%f\n",b,fb);
    }

  }
  printf(" The root is :%f\n",x);
```

Standard Library Functions

Header	**Description**
<stdio.h>	Functions for standard input and output
<float.h>	Floating point size limits
<limits.h>	Contains integral size limits of system
<stdlib.h>	Functions for converting numbers to text and text to numbers, memory allocation, random numbers, other utility functions
<math.h>	Math library functions
<string.h>	String processing functions
<stddef.h>	Common definitions of types used by C

Functions in the library math.h

Function	Returns
sqrt(x)	Square root
exp(x)	Exponential function
log(x)	Natural logarithm (base e)
log10(x)	Logarithm (base 10)
fabs(x)	Absolute value
pow(x,y)	X raised to the power of y
sin(x)	Trignometric sine (x in radians)
cos(x)	Trignometric cosine (x in radians)
tan(x)	Trignometric tangent (x in radians)
atan(x)	Arctangent of x (returned value is in radians)

Using Functions

- Include the header file for the required library using the preprocessor directive

  #include <libraryname.h>

- Note no semi colon after this
- Variables defined in functions are local variables
- Functions have a list of parameters

  Means of communicating information between functions

- Functions can return values
- printf and scanf good examples of function calls
- Use the –lm option to compile an application using math library functions e.g.

  gcc myprog.c –o myprog -lm

Practice

- Build and run the example function1.c
- Add more calls to the blorf() function in the main pprogram
- Build and run function2.c
- Note this avoids the use of the function prototype
- Move the soup function after the main function compile and run what happens?
- Add a prototype and build and run again

Practice

1. Modify the root finding examples for the newton and bisection method to call a function defined by a c- function (rather than inline as performed in the code example)
2. Compile and run functions.c.

   Run the program several times and observe that it always provides. The same output.

   Seed the random number generator using the statement srand(time(NULL));

   Run the program several times and observe the output

Pointers and Arrays

Practice

- Putting it all together
  - Function calls
  - Simple array examples
- Numerical Method Examples
  - Numerical differentiation
  - Numerical Integration

Multidimensional Array

Practical examples

- Compile and run the following programs
  - Numerical Differentiation
    - 2 and four point methods
  - Numerical Integration
    - Trapezium method
    - Simpsons rule (includes lagrange interpolation function)

Practice

- Compile and run the following programs
  - Program array.c initialising and using arrays with pointers
  - Program bubblesort.c is a bubble sort example, using call by reference to manipulate data passed into a function
  - Program arrayref.c uses pointer notation to manipulate arrays

Practice

Modify the integration examples to compute the error function, defined by

$$f(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-x2}$$

Modify the program so that erf(x) is computed for a range of values

Further Sessions

- Building Applications using Make
- From C to C++
- Boost your programming using the standard template libraries