COMsW 1003-1

Introduction to Computer Programming in C

Lecture 14

Spring 2011

Instructor: Michele Merler

http://www1.cs.columbia.edu/~mmerler/comsw1003-1.html

Announcements

Homework 4 out on Wednesday, due o n Monday April 11th

Homework 3 solution out later today

Today

- Midterm Solution
- Finish FILE I/O (from Lecture 13)
- C standard libraries

Midterm Solution

Midterm Solution uploaded to Shared Files in Courseworks

Midterm Statistics

- Average grade: 72
- Standard deviation: 17

C Standard Libraries

- C provides a series of useful functions already implemented in standard libraries
- We have already seen some (stdio.h, string.h)
- In order to use the functions in a library, we must include the library header

```
#include <libraryName.h>
```

C Standard Libraries

C Standard Libraries

- stdio.h : input/output
- string.h : functions on strings
- stdlib.h : utility functions
- math.h : mathematical functions
- ctype.h : character class test
- assert.h : diagnostics
- limits.h and float.h : implementation-defined limits
- time.h : date and time functions
- A few more

C Standard Libraries

- stdio.h : input/output
- string.h : functions on strings
- stdlib.h : utility functions
- math.h : mathematical functions
- ctype.h : character class test
- assert.h : diagnostics
- limits.h and float.h : implementation-defined limits
- time.h : date and time functions
- A few more

testLibraries.c

stdio.h

- Standard input and output
- Input/output from command line (keyborad)
  - `fprintf(), fgets(), sscanf()`
- Input/output from files
  - `FILE, fopen(), fclose()`

testLibraries.c

string.h

Operations involving strings

```
string s1, s2;
char c;
```

- `int n = strcmp( s1, s2)` : compare s1 and s2, if(s1==s2) -> n = 0
- `int len = strlen(s1)` : return length of s1
- `char *pc = strchr(s1, c)`: return pointer to first occurrence of c in s1
- `char *ps = strstr(s1, s2)`: return pointer to first occurrence of string s2 in s1, or NULL if not present
- `char *strcpy(s1, s2)` : copy string s2 into s1, return s1
- `char *strcat(s1, s2)` : append s2 to s1 (concatenate), return s1
- `char *strtok(s1, s2)` : split long strings into pieces, or tokens

testLibraries.c

stdlib.h

Number conversions

- `float nf = atof(const char *s)` : converts string s to float
- `int n = atoi(const char * s)` : convert string s to int

Memory allocation

`malloc(), free()` : memory management

Other utilities

- `int n = rand()` : returns a (pseudo) random int between 0 and constant RAND_MAX
- `void srand(unsigned int n)` : seeds rand generator
- `system(string s)` : runs ***s*** in OS

testLibraries.c

math.h

- Mathematical functions
- Often needs to be specially linked when compiling because takes advantage of specialized math hardware in processor

gcc **-lm** –Wall -o myProgram myProgram.c

```
double functionName(  double c )
```

- `sin(x), cos(x), tan(x)`
- `exp(x), log(x), log10(x)` : e^x , natural and base-10 logarithm
- `pow(x,y)` : x^y
- `sqrt(x)` : square root
- `ceil(x), floor(x)` : closest int above or below
- `y = fabs(x)` : absolute value , if x = -3.2, y will be 3.2

ctype.h

testLibraries.c

Utility functions to check for types of char

```
int functionName(  unsigned char c )
```

- `isalpha(c)` : check if c is an alphabet character ‘a’-’z’, ‘A’-’Z’
- `isdigit(c)` : check if c is digit ‘0’-’9’
- `isalnum(c)` : isalpha(c) or isdigit(c)
- `iscntrl(c)` : control char (i.e. \n, \t, \b)
- `islower(c)` , `isupper(c)` : lowercase/uppercase

Return value is 0 if false , != 0 if true

testLibraries.c

ctype.h

Utility functions to convert from lower case to upper case

```
char functionName(char c )
```

- `d = tolower(c)` : if c is ‘T’, d will be ‘t’
- `d = toupper(c)` : if c is ‘m’ , d will be ‘M’

limits.h and float.h

testLibraries.c

Contain various important constants such as the minimum and maximum possible values for certain types, sizes of types, etc.

- CHAR_BIT (bits in a char)
- INT_MAX, CHAR_MAX, LONG_MAX

  (maximum value of int, char, long int)
- INT_MIN, CHAR_MIN, LONG_MIN
- FLT_DIG (decimal digits of precision)
- FLT_MIN, FLT_MAX (min. and max. value of float)
- DBL_MIN, DBL_MAX (and of double precision float)

time.h

testLibraries.c

Provides new **type** to represent time, `time_t`

- `time_t time(NULL)` : returns current time
- `time_t clock()` : returns processor time used by program since beginning of execution
- strftime(A, sizeof(A), “formatted text”, time struct) :

  format text with placeholders:

  %a weekday
  %b month
  %c date and time
  %d day of month
  %H hour

testLibraries.c

assert.h

- Provides a macro to check if critical conditions are met during your program
- Nice way to test programs

```
assert( expression )
```

If the expression is false, the program will print to command line:

Assertion failed: *expression* , file *filename* , line *lineNumber*

More

- **stdarg.h** : allows you to create functions with variable argument lists
- **signal.h** - provides constants and utilities for standardized error codes for when things go wrong