NWEN 241
C Fundamentals

Qiang Fu

School of Engineering and Computer Science
Victoria University of Wellington

Victoria
UNIVERSITY OF WELLINGTON
Te Whare Wānanga
o te Ūpoko o te Ika a Māui
CAPITAL CITY UNIVERSITY

This Lecture

- Background about C
- Development environment & C program structures

Comparing C, C++, Java

- The C Family of Languages: Interview with Dennis Ritchie, Bjarne Stroustrup, and James Gosling:
  – http://www.gotw.ca/publications/c_family_interview.htm

Comparing C, C++ and Java

- C is the basis for C++ and Java
  – C evolved into C++
  – C++ transmuted into Java
  – The “class” is an extension of “struct” in C
- Similarities
  – Java uses a syntax similar to C++ (for, while, …)
  – Java supports OOP as C++ does (class, inheritance, …)
- Differences
  – Java does not support pointer
  – Java frees memory by garbage collection
  – Java is more portable by using bytecode and virtual machine
  – Java does not support operator overloading
  – ….

Background and Characteristics

- Designed by Dennis Ritchie of Bell Labs in the 1970s
- An outgrowth of B also developed at Bell Labs
- ANSI/ISO standard in early 1990s.
- Bridging the gap between machine language and high-level languages
  - Low-level features: fast/efficient (systems programming)
  - High-level features: structured programming (applications programming)

Applications

- Operating systems
- Distributed systems
- Network programming
- Database applications
- Real-time and engineering applications
- Any application where efficiency is *paramount*

Development Environment

- Lab: CO246
- ID access cards (Swipe Cards): should work if you are registered in NWEN 241

- PC Unix workstations, Linux, KDE
- Network file system
- Tools: gcc, g++, gdb, eclipse, emacs, gedit, vi, vim
- Text editor vs IDE: text editor recommended

- Remote access: https://ecs.victoria.ac.nz/Support/TechNoteWorkingFromHome

Program Structure

- A C program consists of one or more *functions*
- A C program must have a `main` function

```
int main(void)
{ ...;
  return 0;
}
```

- Execution begins with the `main` function
- Java vs. C
  - C uses stand-alone functions
  - No stand-alone functions in Java
  - No global functions in Java

Program Structure

- Each function must contain:
  - A function *heading*, (return type, function name, an *optional* list of *arguments*)
  - A list of argument *declarations*, if arguments are included in heading
  - A *compound statement*

```
int function_name(int x, int y)
{
  ...
}
```

Program Structure

- An example (single function)

```
/* A simple program */            /* comment */

#include <stdio.h>                /* library file access */

int main(void)                    /* function heading  */
{
   printf("Hello world\n");       /* output statement */

   return 0;                       /* return statement */
}
```

Program Structure

- An example (single function)

```
/* Program to calculate the area of a circle */   /* comment */

#include <stdio.h>                /* library file access */
#define PI 3.14       /* macro definition – symbolic constant */
#define SQ(x) ((x)*(x))           /* macro with arguments */

int main(void)                    /* function heading  */
{
   float radius, area;            /* variable declarations */

   printf("Radius = ");           /* output statement (prompt)*/
   scanf("%f", &radius);          /* input statement */

   area = 3.14 * radius * radius; /* assignment statement */
   printf("Area1 = %f\n", area);  /* output statement */

   area = PI * SQ(radius);        /* use macros */
   printf("Area2 = %f\n", area);  /* output statement */

   return 0;                      /* return statement */
}
```

Program Structure

- Another example (multiple functions)

```
/* Program to calculate the area of a circle */

#include <stdio.h>                       /* library file access */
#define PI 3.1415926            /* macro definition - symbolic constant */

float sq(float);                /* square function - function prototype */

int main(void)                           /* function heading  */
{
   float radius, area;                   /* variable declarations */

   printf("Radius = ");                  /* output statement (prompt)*/
   scanf("%f", &radius);                 /* input statement */

   area = PI * sq(radius);               /* use square function */
   printf("Area = %f\n", area);          /* output statement */
   return 0;                             /* return statement */
}

float sq(float r)
{  return (r * r);}             /* square function - function definition*/
```

Summary

- C / C++ / Java
- C program structure

Next Lecture

- More on C fundamentals