Computer Programming using C
Lecture 6

Dr Julian Miller
Room P/M/101
Ext 2383; E-mail: jfm@ohm
http://www.elec.york.ac.uk/intsys/users/jfm7/

Introduction

• Revision (in the lab last week)
• Answers to exercises
• In the lab this week
In the lab two weeks ago (lab 4)

• The C Preprocessor
  – Allows the user to affect the code before it is compiled
  – All preprocessor instructions begin with #
  – Useful mainly for
    • Debugging
    • Clarity of code

#include

• #include <stdio.h>
  Or
• #include “myfile”
Text substitution with `#define`

- `#define`  **TAG**  **REPLACEMENT**

```c
#include <stdio.h>
#define HELLO_MESSAGE "Hello, World!\n"
#define PI 3.14159265

int main(void) {
    printf(HELLO_MESSAGE);
    printf("Pi = %lf (roughly)\n", PI);
    return 0;
}
```

Conditional compilation  
`#if defined`

```c
#include <stdio.h>
#define DEBUG

int main(void) {
    /* Declare some variables for the calculations */
    int distance_to_tokyo, distance_to_airport;
    int speed_of_plan, speed_of_car;
    int time_to_fly, time_to_driver, time_to_tokyo;
    int average_speed;
    distance_to_tokyo = 9725;
    distance_to_airport = 115;
    speed_of_plan = 22000;
    speed_of_car = 100;
    time_to_fly = (distance_to_tokyo - distance_to_airport) / speed_of_plan;
    time_to_drive = distance_to_airport / speed_of_car;
    if defined(DEBUG)
        printf("time_to_fly = %d, time_to_driver = %d\n", time_to_fly, time_to_driver);
    #endif

```

```c
average_speed = distance_to_tokyo / time_to_tokyo;
printf("It takes %d hours to get to Tokyo\n", time_to_tokyo);
printf("The average speed would be %d mph\n", average_speed);
return 0;
```
In the lab last week (lab 5):  
Functions

- Functions are parts of a program that can be run by other parts of a program
- You have used predefined functions (e.g. printf, scanf)
- Functions look like this
  
  ```c
  type function_name(parameters)
  ```
- Where ‘type’ means a C data type
  - int, char, double, unsigned, void
Example program

```c
#include <stdio.h>
void display_welcome(void)
{
    printf("welcome\n");
}
int square(int number)
{
    return number * number;
}
int main(void)
{
    int value;
    display_welcome();
    printf("Enter an integer: ");
    scanf("%d", &value);
    printf("The square of %d is %d\n", value, square(value));
    return 0;
}
```

What happens when this is compiled?

```c
#include <stdio.h>
void display_welcome(void)
{
    printf("welcome\n");
}
int square(int number)
{
    return number * number;
}
int main(void)
{
    int value;
    double value1;
    display_welcome();
    printf("Enter an integer: ");
    scanf("%d", &value);
    value1 = value;
    printf("The square of %d is %d\n", value, square(value1));
    return 0;
}
```
Mathematical functions: math.h

- In math.h there are many mathematical functions that you can use:
  - sqrt, pow, sin, cos, asin, ...
  - They all return doubles and take doubles as arguments, however you can still use them with int arguments as they get converted to doubles.

Exercise 5.9: stick man program with functions

```c
int main(void)
{
    int x_head, y_head, x_ground, y_ground;
    int go, num_goes = 3;
    printf("Welcome to the stick man game\n\n");
    for (go = 0; go < num_goes; go++)
    {
        printf("You are on go %d\n", go+1);
        x_head = 100;
        y_head = 200;
        x_ground = 0;
        initwindow(X_WINDOW, Y_WINDOW);
        y_ground = draw_ground(x_ground, y_head, WHITE);
        draw_stick_man(x_head, y_head, RED);
        x_head = move_stick_man(x_head, y_head, RED);
        plot_projectile(x_head, y_head, y_ground, BLUE);
        printf("In GRAPHICS window Press a key to end\n");
        getkey();
    }
    closegraph();
    return 0;
}
```
Stick man program: #statements

```
#define PI 3.141592653589
#define GRAVITY 9.8
#define RETURN 13
#define LEFTARROW 75
#define RIGHTARROW 77

/* drawing area constants */
#define X_WINDOW 640
#define Y_WINDOW 480

/* stick man constants */
#define HEADSIZE 20
#define ARMLENGTH 20
#define BODYSIZE 80
#define ARMPOSITION 40
#define STRIDE 30
#define LEG_LENGTH 40
#define STICK_MAN_HEIGHT HEADSIZE+BODYSIZE+LEG_LENGTH

#include "graphics_lib.h"
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
```

Stick man: function prototypes

```
int draw_ground(int x_ground, int y_head, int COLOUR);
void draw_stick_man(int x_head, int y_head, int COLOUR);
int move_stick_man(int x_head, int y_head, int COLOUR);
void plot_projectile(int x_head, int y_head, int y_ground, int COLOUR);
```
Stick man: draw ground

```c
int draw_ground(int x_ground, int y_head, int COLOUR)
{
    int y_ground;
    y_ground=y_head+BODY_SIZE+LEG_LENGTH+3;
    setcolor(COLOUR);
    line(x_ground,y_ground,X_WINDOW,y_ground);
    return y_ground;
}
```

Stick man: draw stick man

```c
void draw_stick_man(int x_head, int y_head, int COLOUR)
{
    setcolor(COLOUR);
    /* draw head */
    circle(x_head, y_head, HEAD_SIZE);
    /* Draw Body */
    line(x_head,y_head+HEAD_SIZE,x_head,y_head+BODY_SIZE);
    /* Draw arms */
    line(x_head-ARM_LENGTH,y_head+ARMPOSITION,x_head+ARM_LENGTH,y_head+ARMPOSITION);
    /* draw legs */
    line(x_head-STRIDE,y_head+BODY_SIZE+LEG_LENGTH,x_head,y_head+BODY_SIZE);
    line(x_head+STRIDE,y_head+BODY_SIZE+LEG_LENGTH,x_head,y_head+BODY_SIZE);
}
```
Input buffer

• You also learned that when things are typed in at the keyboard they go into something called the ‘input buffer’
• `scanf` tries to match the format string with the contents of the buffer
• If there is no match things are left in the buffer and often you need to clear the buffer with
  `fflush(stdin);`

In the lab this week (lab 6)

• Arrays and strings
  – Arrays are a very powerful feature of programming languages. They allow many pieces of information to be stored and altered within programs and functions
  – Strings are arrays of characters (i.e. words). `scanf` and `printf` have a special format specifier for these, `%s`