CS2100 COMPUTER ORGANISATION

http://www.comp.nus.edu.sg/~cs2100/

Lecture #4c

Pointers and Functions

Questions?

IMPORTANT: DO NOT SCAN THE QR CODE IN THE VIDEO RECORDINGS. THEY NO LONGER WORK

Ask at
https://sets.netlify.app/module/676ca3a07d7f5ffc1741dc65

OR

Scan and ask your questions here!
(May be obscured in some slides)

2. Calling Functions (1/3)

- In C, there are many libraries offering functions for you to use.
- Eg: scanf() and printf() – requires to include <stdio.h>
- C provides many libraries, for example, the math library
- To use math functions, you need to
  - Include <math.h> AND
  - Compile your program with –lm option (i.e. gcc –lm …) in sunfire
- See table (next slide) for some math functions

2. Calling Functions (2/3)

Function	Arguments	Result
abs(x)	int	int
ceil(x)	double	double
cos(x)	double (radians)	double
exp(x)	double	double
fabs(x)	double	double
floor(x)	double	double
log(x)	double	double
log10(x)	double	double
ceil(x)	double	double
pow(x, y)	double, double	double
sin(x)	double (radians)	double
sqrt(x)	double	double
tan(x)	double (radians)	double

Function prototype:

double pow(double x, double y)

function return type

2. Calling Functions (3/3)

To link to Math library

MathFunctions.c

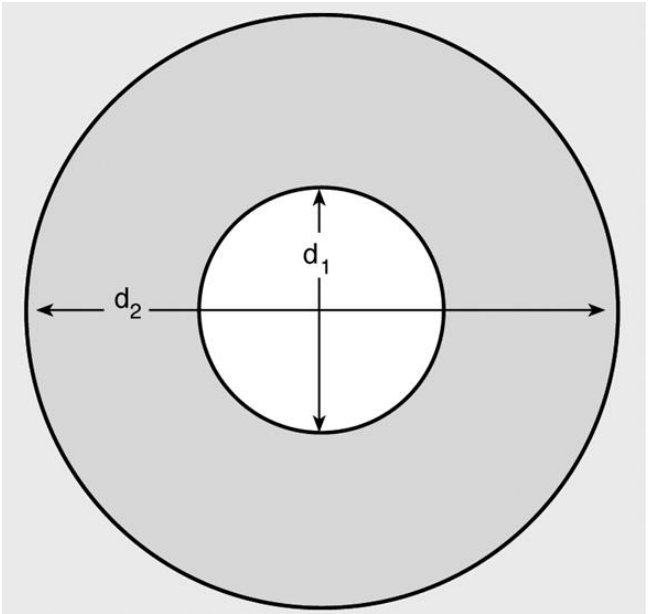
```
#include <stdio.h>
#include <math.h>

int main(void) {
   int   x, y;
   float val;

   printf("Enter x and y: ");
   scanf("%d %d", &x, &y);
   printf("pow(%d, %d) = %f\n", x, y, pow(x,y));

   printf("Enter value: ");
   scanf("%f", &val);
   printf("sqrt(%f) = %f\n", val, sqrt(val));

   return 0;
}
```

```
$ gcc -lm MathFunctions.c
$ a.out
Enter x and y: 3 4
pow(3,4) = 81.000000
Enter value: 65.4
sqrt(65.400002) = 8.087027
```

3. User-Defined Functions (1/6)

- We can define and use our own functions

Example: Compute the volume of a flat washer. Dimensions of a flat washer are usually given as an inner diameter, an outer diameter, and a thickness.



$$rim\ area = \pi(d_2/2)^2 - \pi(d_1/2)^2$$

3. User-Defined Functions (2/6)

Washer.c

```
#include <stdio.h>
#include <math.h>
#define PI 3.14159
int main(void) {
    double d1, // inner diameter
           d2, // outer diameter
           thickness, outer_area, inner_area, rim_area, volume;

    // read input data
    printf("Enter inner diameter, outer diameter, thickness: ");
    scanf("%lf %lf %lf", &d1, &d2, &thickness);

    // compute volume of a washer
    outer_area = PI * pow(d2/2, 2);
    inner_area = PI * pow(d1/2, 2);
    rim_area = outer_area - inner_area;
    volume = rim_area * thickness;

    printf("Volume of washer = %.2f\n", volume);
    return 0;
}
```

```
Enter …: 8.2 10.5 2.2
Volume of washer = 74.32
```

3. User-Defined Functions (3/6)

- Note that area of circle is computed twice. For code reusability, it is better to define a function to compute area of a circle.

```
double circle_area(double diameter) {
   return PI * pow(diameter/2, 2);
}
```

- We can then call/invoke this function whenever we need it.

`circle_area(d2)` → to compute area of circle with diameter d2

`circle_area(d1)` → to compute area of circle with diameter d1

3. User-Defined Functions (4/6)

WasherV2.c

```
#include <stdio.h>
#include <math.h>
#define PI 3.14159

double circle_area(double);

int main(void) {
    // code similar to Washer.c; omitted here

    // compute volume of a washer
    rim_area = circle_area(d2) - circle_area(d1);
    volume = rim_area * thickness;

    printf("Volume of washer = %.2f\n", volume);
    return 0;
}

// This function returns the area of a circle
double circle_area(double diameter) {
    return PI * pow(diameter/2, 2);
}
```

Function prototype

Function definition

3. User-Defined Functions (5/6)

- It is a good practice to put function prototypes at the top of the program, before the main() function, to inform the compiler of the functions that your program may use and their return types and parameter types.
- A function prototype includes only the function's return type, the function's name, and the data types of the parameters (names of parameters are optional).
- Function definitions to follow after the main() function.
- Without function prototypes, you will get error/warning messages from the compiler.

3. User-Defined Functions (6/6)

- Let's remove (or comment off) the function prototype for circle_area() in WashersV2.c
- Messages from compiler:

```
WashersV2.c: In function 'main':
WashersV2.c:19:2: warning: implicit declaration of function
'circle_area' [-Wimplicit-function-declaration]
  rim_area = circle_area(d2) - circle_area(d1);
  ^
WasherV2.c: At top level:
WashersV2.c:27:8: error: conflicting types for 'circle-area'
:
```

- Without function prototype, compiler assumes the default (implicit) return type of int for circle_area() when the function is used in line 19, which conflicts with the function header of circle_area() when the compiler encounters the function definition later in line 27.

End of File