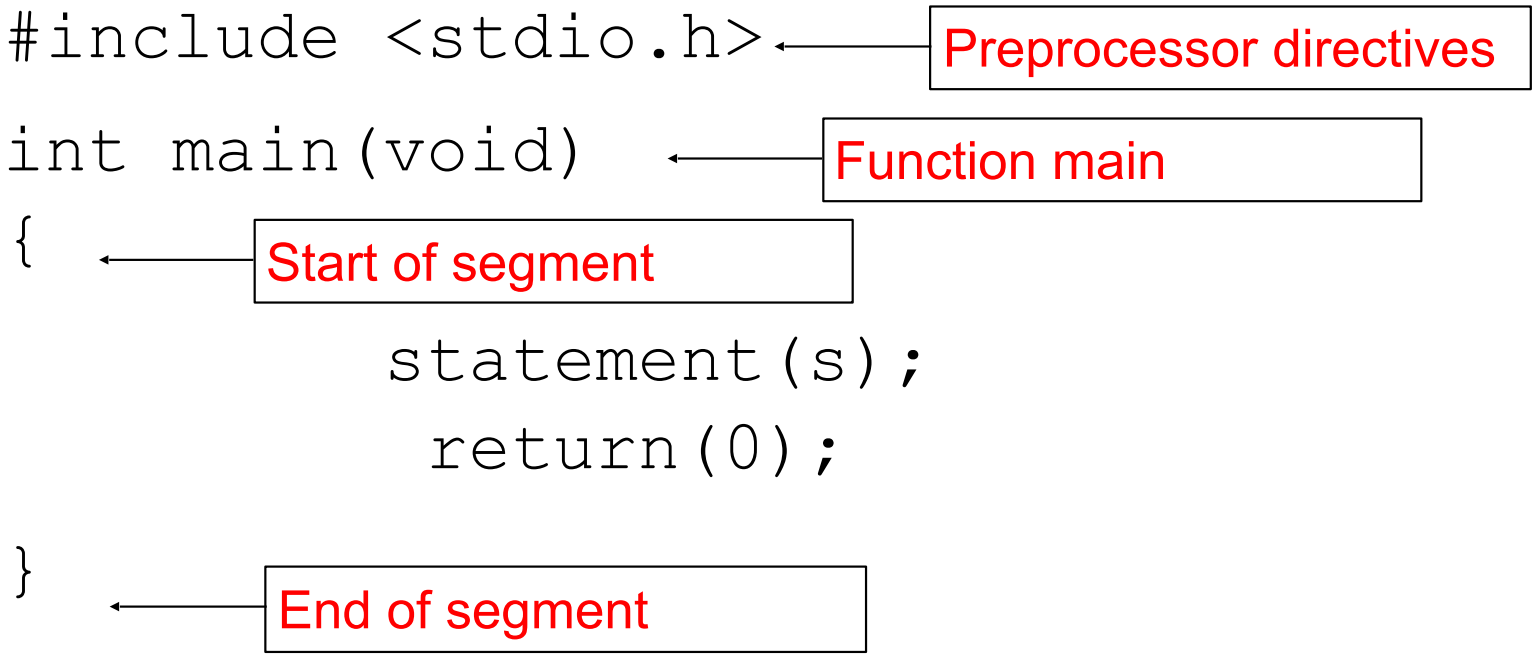
I/O Issues in C

Robbie

CSCI2100 Data Structures Tutorial 2

C program skeleton

- In short, the basic skeleton of a C program looks like this:



Input/Output Operations

- Input operation
  - an instruction that copies data from an input device into memory
- Output operation
  - an instruction that displays information stored in memory to the output devices (such as the monitor screen)

Input/Output Functions

- A C function that performs an input or output operation
- A few functions that are pre-defined in the header file <stdio.h> such as :
  - printf()
  - scanf()
  - getchar(), putchar()

The printf() function

- Used to send data to the standard output (usually the monitor) to be printed according to specific format.
- General format:
  - `printf("string literal");`
    - A sequence of any number of characters surrounded by double quotation marks.
  - `printf("format string", variables);`
    - Format string is a combination of text, conversion specifier and escape sequence.

The printf() function cont…

- Example:
  - `printf("Thank you\n");`

```
Thank you
```

  - `printf ("Total sum is: %d\n", sum);`

```
Total sum is: 50
```

Assuming that the value of sum is 50

- %d is a placeholder (conversion specifier)
  - marks the display position for a type integer variable
  - Common Conversion Identifier used in printf function
- \n is an escape sequence
  - moves the cursor to the new line

	printf
int	%d
float	%f
double	%f
char	%c
string	%s

Escape Sequence

Escape Sequence	Effect
\a	Beep sound
\b	Backspace
\f	Formfeed (for printing)
\n	New line
\r	Carriage return
\t	Tab
\v	Vertical tab
\\	Backslash
\”	“ sign
\o	Octal decimal
\x	Hexadecimal
\0	NULL

Placeholder/Conversion Specifier

No	Conversion Specifier	Output Type	Output Example
1	%d	Signed decimal integer	76
2	%i	Signed decimal integer	76
3	%o	Unsigned octal integer	134
4	%u	Unsigned decimal integer	76
5	%x	Unsigned hexadecimal (small letter)	9c
6	%X	Unsigned hexadecimal (capital letter)	9C
7	%f	Integer including decimal point	76.0000
8	%e	Signed floating point (using e notation)	7.6000e+01
9	%E	Signed floating point (using E notation)	7.6000E+01
10	%g	The shorter between %f and %e	76
11	%G	The shorter between %f and %E	76
12	%c	Character	‘7’
13	%s	String	‘76'

The scanf() function

- Read data from the standard input device (usually keyboard) and store it in a variable.
- General format:
  - `scanf("format string", &variable);`
- Notice ampersand (&) operator :
  - C address of operator
  - it passes the address of the variable instead of the variable itself
  - tells the scanf() where to find the variable to store the new value
- Format string is a combination of conversion specifier and escape sequence (if any).

The scanf() function cont…

- Common Conversion Identifier used in printf and scanf functions.

	printf	scanf
int	%d	%d
float	%f	%f
double	%f	%lf
char	%c	%c
string	%s	%s

- Example :

```
int age;
printf("Enter your age:");
scanf("%d", &age);
```

The scanf() function cont…

- If you want the user to enter more than one value, you serialize the inputs.
- Example:

```
float height, weight;

printf(“Please enter your height and weight:”);
scanf(“%f%f”, &height, &weight);
```

getchar() and putchar()

- getchar() - read a character from standard input
- putchar() - write a character to standard output
- Example:

```
Please type a character: h
You have typed this character: h
```

```
#include <stdio.h>
int main(void)
{
   char my_char;
   printf("Please type a character:");
   my_char = getchar();
   printf("You have typed this character: ");
   putchar(my_char);
   return (0);
}
```

getchar() and putchar() cont

- Alternatively, you can write the previous code using normal printf / scanf and %c placeholder.
- Example:

```
Please type a character: h
You have typed this character: h
```

```
#include <stdio.h>
int main(void)
{
  char my_char;
  printf("Please type a character: ");
  scanf("%c",&my_char);
  printf("You have typed this character: %c", my_char);
  return(0);
}
```

Constants

- Character constants
  - A character enclosed in a single quotation mark
  - Example:
    - `const char letter = 'n';`
    - `const char number = '1';`
    - `printf("%c", 'S');`
- Enumeration
  - Values are given as a list
  - Example:

```
enum Language {
    Malay,
    English,
    Arabic
};
```

Constant example – volume of a cone

```
#include <stdio.h>

int main(void)
{
   const double pi = 3.412;
   double height, radius, base, volume;

   printf("Enter the height and radius of the cone:");
   scanf("%lf %lf", &height, &radius);

   base = pi * radius * radius;
   volume = (1.0/3.0) * base * height;

   printf("The volume of a cone is %f ", volume);
   return (0);
}
```

#define

```
#include <stdio.h>
#define pi 3.142

int main(void)
{
   double height, radius, base, volume;

   printf("Enter the height and radius of the
   cone:");
   scanf("%lf %lf", &height, &radius);

   base = pi * radius * radius;
   volume = (1.0/3.0) * base * height;

   printf("The volume of a cone is %f ", volume);
   return (0);
}
```

String Literal

- A sequence of any number of characters surrounded by double quotation marks “ ”.
- Example of usage in C program:

```
printf(“What a beautiful day.\n”);
```

```
What a beautiful day.
```

- To have double quotation marks as part of the sentence, precede the quote with backslash

```
printf(“He shouted \“stop!\” to the thief.\n”);
```

```
He shouted “stop!” to the thief.
```

Sample Problem 1: Division

- Given n pair of integers, for each pair, please calculate the quotient and remainder of a/b followed by the exact real number result of a/b (to the nearest hundredth)
- Sample input:

2

4 6

3 2

- Sample output:

0 4 0.67

1 1 1.50

Sample Problem 2: String reversion

- Given n strings (no white space in all strings), for each string please reverse it.
- Sample input:

3
asdfghjkl123
qwertyuiop456
zxcvbnm789

- Sample output:

321lkjhgfdsa
654poiuytrewq
987mnbvcxz