

CSE1301 Computer Programming Lecture 7: Input/Output

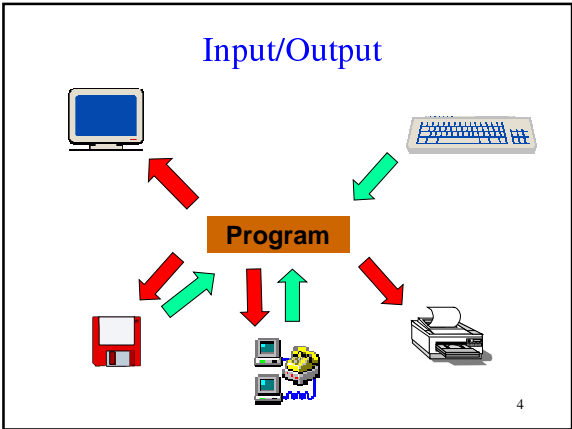
1

- ## Topics
- Streams
 - Formatted input
 - Formatted output
- 2

Recall

- `scanf()`
Example:
`scanf("%d", &x);`
- `printf()`
Example:
`printf("The value of x is %d\n", x);`
- `#include <stdio.h>`

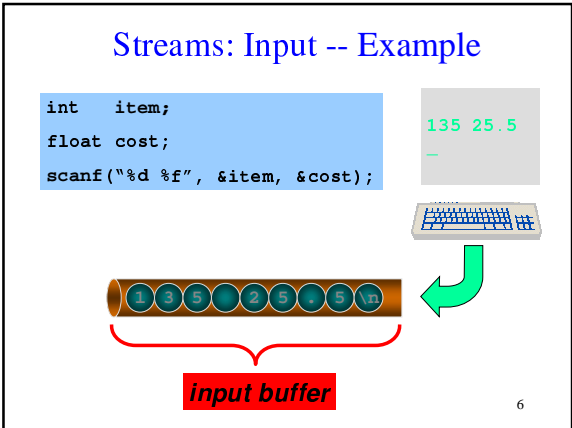
3



Streams

- Text input or output is dealt with as a sequence of characters
- A stream serves as a channel to convey characters between I/O and programs

5



Streams: Input -- Example (cont)

```
int item;
float cost;
scanf("%d %f", &item, &cost);
```

135 25.5

7

Streams: Input -- Example (cont)

```
int item;
float cost;
scanf("%d %f", &item, &cost);
```

135 25.5

8

Streams: Input -- Example (cont)

```
int item;
float cost;
scanf("%d %f", &item, &cost);
```

135 25.5

9

Streams: Output -- Example

```
printf("Hello!\n");
```

10

Streams: Output -- Example (cont)

```
printf("Hello!\n");
```

11

Streams: Output -- Example (cont)

```
printf("Hello!\n");
```

12

Streams: Output – Example (cont)

```
printf("Hello!\n");
```

13

Streams: Output – Example (cont)

```
printf("Hello!\n");
```

14

Streams: Output – Example (cont)

```
printf("Hello!\n");
```

15

Streams: Output – Example (cont)

```
printf("Hello!\n");
```

16

Streams: Output – Example (cont)

```
printf("Hello!\n");
```

17

Streams: Output – Example (cont)

```
printf("Hello!\n");
```

18

Streams

- From the program's point of view, the characters are queued in a pipe
- The sequence of characters is organized into lines
- Each line:
 - can have zero or more characters
 - ends with the "newline" character `'\n'`

19

"Standard" Streams

- Standard streams :
 - `stdin` - standard input
 - usually from keyboard
 - `stdout` - standard output
 - usually to screen
 - `stderr` - standard error
 - usually to screen
- must have at the top of your program `#include <stdio.h>`
- can be *redirected*

20

`stdin`: Input

- Data is read in from `stdin` (into a variable) using the `scanf()` function
- When input ends, the `scanf()` function returns a special value: `EOF`

21

Example: ReadData

Input name, age, gender, idNumber

22

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

23

```
#include <stdio.h>
/*****\
Read in important info about a student
*****/
```

24

```
#include <stdio.h>
/*****\
Read in important info about a student
\*****/
int main()
{

return 0;
}
```

25

```
#include <stdio.h>
/*****\
Read in important info about a student
\*****/
int main()
{
char name[100] ;
float age ;
char gender ;
int idNumber ;

return 0;
}
```

26

```
#include <stdio.h>
/*****\
Read in important info about a student
\*****/
int main()
{
char name[100] ;
float age ;
char gender ;
int idNumber ;

scanf("%s %f %c %d", name, &age, &gender, &idNumber);
return 0;
}
```

27

```
#include <stdio.h>
/*****\
Read in important info about a student
\*****/
int main()
{
char name[100] ;
float age ;
char gender ;
int idNumber ;

scanf("%s %f %c %d", name, &age, &gender, &idNumber);
return 0;
}
```

Input: Ashley 19.2 M 3825

28

stdout : Output

- Data (e.g., from a variable) is written out to **stdout** using the **printf()** function.

29

Example: WriteData

Set name to "Ashley"
 Set age to 18.2
 Set gender to 'M'
 Set idNumber to 3825
 Output name, age, gender, idNumber

30

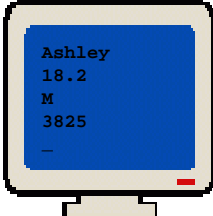
```

#include <stdio.h>

/*****
Write out important info about a student
*****/

int main()
{
    char *name = "Ashley";
    float age = 18.2;
    char gender = 'M';
    int idNumber = 3825;

    printf("%s\n%f\n%c\n%d\n", name, age, gender, idNumber);
    return 0;
}
    
```



31

Formatted Input and Output

- General form:


```
printf(format-control-string, other-arguments);
```

```
scanf(format-control-string, other-arguments);
```
- Examples:


```
printf("%s\n%f\n%c\n%d\n", name, age, gender, idNumber);
```

```
scanf("%s %f %c %d", name, &age, &gender, &idNumber);
```

32

printf -- Format-Control-String

- Describes the format of the data for output
- Contains “conversion specifiers” and “literal characters”

Example:

```
printf("%s is %d years old.\n", name, age);
```

33

printf -- Format-Control-String (cont)

- Describes the format of the data for output
- Contains “conversion specifiers” and “literal characters”

Example:

```
printf("%s is %d years old.\n", name, age);
```

↑ ↑
conversion specifiers

34

printf -- Format-Control-String (cont)

- Describes the format of the data for output
- Contains “conversion specifiers” and “literal characters”

Example:

```
printf("%s is %d years old.\n", name, age);
```

↑ ↑
literal characters

35

printf -- Other-Arguments

- For printf: variables containing data for output

Example:

```
printf("%s is %d years old.\n", name, age);
```

↑ ↑

36

scanf -- Format-Control-String

- Describes the format of the data given as input
- Contains “conversion specifiers”

Example:

```
scanf("%s %f %c %d", name, &age, &gender, &id);
```

37

scanf -- Other-Arguments

- For `scanf`: “pointers” to variables where the input will be stored

Example:

```
scanf("%s %f %c %d", name, &age, &gender, &id);
```

38

scanf -- Other-Arguments (cont)

- For `scanf`: “pointers” to variables in which the input will be stored

Example:

```
scanf("%s %f %c %d", name, &age, &gender, &id);
```

- Variables of type `int`, `float` or `char` need ‘&’
- Do NOT use ‘&’ with strings!
- ‘&’ is for `scanf` only!

39

Common Conversion Specifiers for Numerical Information

- decimal integer: `%d`

```
printf("What is %d plus %d?\n", x, y);
scanf("%d", &sum);
```
- float: `%f`

```
printf("%f squared is...? ", x);
scanf("%f", &ans);
```
- double:

```
printf("%f squared is...? ", x);
scanf("%lf", &ans);
```

40

Conversion Specifiers for Alphanumeric Information

- char: `%c`

```
printf("What letter follows %c?\n", ch);
scanf("%c", &nextchar);
```
- string: `%s`

```
printf("Name: %s\n", name);
scanf("%s", name);
```

41

printf: Conversion Specifiers

- `i` or `d`: display a signed decimal integer
- `f`: display a floating point value
- `e` or `E`: display a floating point value in exponential notation
- `g` or `G`: display a floating point value in either `f` form or `e` form
- `L`: placed before any float conversion specifier to indicate that a long double is displayed

42

scanf: Conversion Specifiers

- **d**: read an optionally signed decimal integer
- **i**: read an optionally signed decimal, octal, or hexadecimal integer

i and **d**: the argument is a "pointer" to an integer

```
int idNumber;
scanf("%d", &idNumber);
```

43

scanf: Conversion Specifiers (cont)

- **h** or **l**: placed before any integer conversion specifiers to indicate that a short or long integer is to be input

```
long int idNumber;
scanf("%ld", &idNumber);
```
- **l** or **L**: placed before any float conversion specifiers to indicate that a double or long double is to be input

44

Conversion Example

Input octal integer

Output integer as decimal

45

Conversion Example (cont)

```
#include <stdio.h>
int main()
{
    int i ;

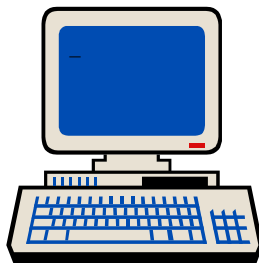
    scanf("%o", &i);
    printf("%d\n", i);
    return 0;
}
```

46

Conversion Example (cont)

```
#include <stdio.h>
int main()
{
    int i ;

    scanf("%o", &i);
    printf("%d\n", i);
    return 0;
}
```

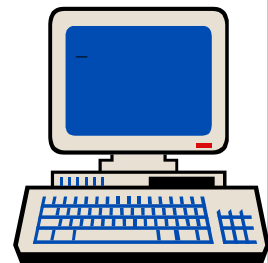


47

Conversion Example (cont)

```
#include <stdio.h>
int main()
{
    int i ;

    scanf("%o", &i);
    printf("%d\n", i);
    return 0;
}
```

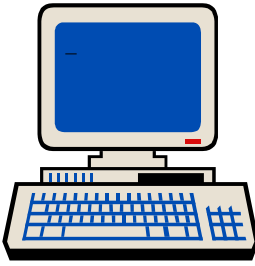


48


Conversion Example (cont)

```
#include <stdio.h>
int main()
{
    int i ;

    scanf("%o", &i);
    printf("%d\n", i);
    return 0;
}
```



i

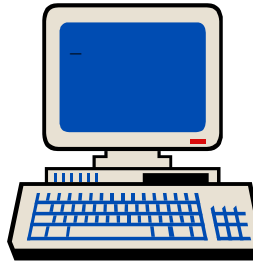


49

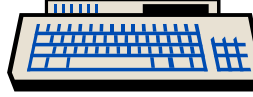
Conversion Example (cont)

```
#include <stdio.h>
int main()
{
    int i ;

    scanf("%o", &i);
    printf("%d\n", i);
    return 0;
}
```



i

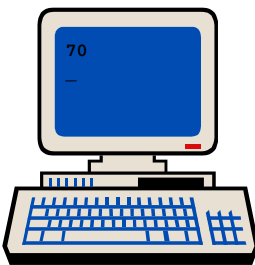


50


Conversion Example (cont)

```
#include <stdio.h>
int main()
{
    int i ;

    scanf("%o", &i);
    printf("%d\n", i);
    return 0;
}
```



i

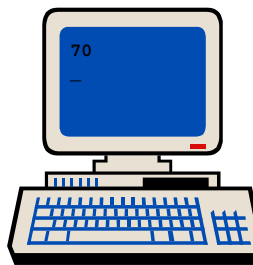


51


Conversion Example (cont)

```
#include <stdio.h>
int main()
{
    int i ;

    scanf("%o", &i);
    printf("%d\n", i);
    return 0;
}
```



i

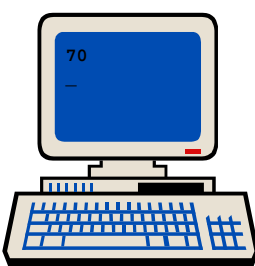


52

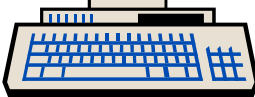
Conversion Example (cont)

```
#include <stdio.h>
int main()
{
    int i ;

    scanf("%o", &i);
    printf("%d\n", i);
    return 0;
}
```



i

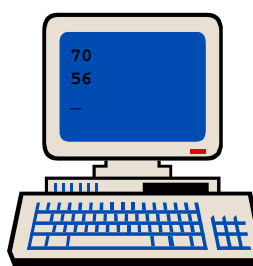


53

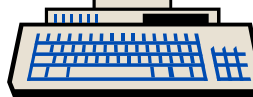
Conversion Example (cont)

```
#include <stdio.h>
int main()
{
    int i ;

    scanf("%o", &i);
    printf("%d\n", i);
    return 0;
}
```



i



54

Skipping Characters in Input Stream

- Skipping blank spaces

```
scanf("%d %d %d", &day, &month, &year);
```

- Skipping dashes

- Enter data as dd-mm-yyyy: 16-3-1999
- Store each number in date variables

```
scanf("%d-%d-%d", &day, &month, &year);
```

55

Summary

- **Input** from keyboard is via the **stdin** stream
- **Output** to the screen is via the **stdout** stream
- Streams carry characters
 - divided into lines with '\n' character
 - input ends with special value: **EOF**
- To use the C I/O functions, you must include the **stdio.h** header file
- Input and output can be formatted and converted between data types

56

Reading

- King: Chapters 3 and 7, Section 22.3
- Deitel & Deitel: Sections 9.1 to 9.6, 9.11

57