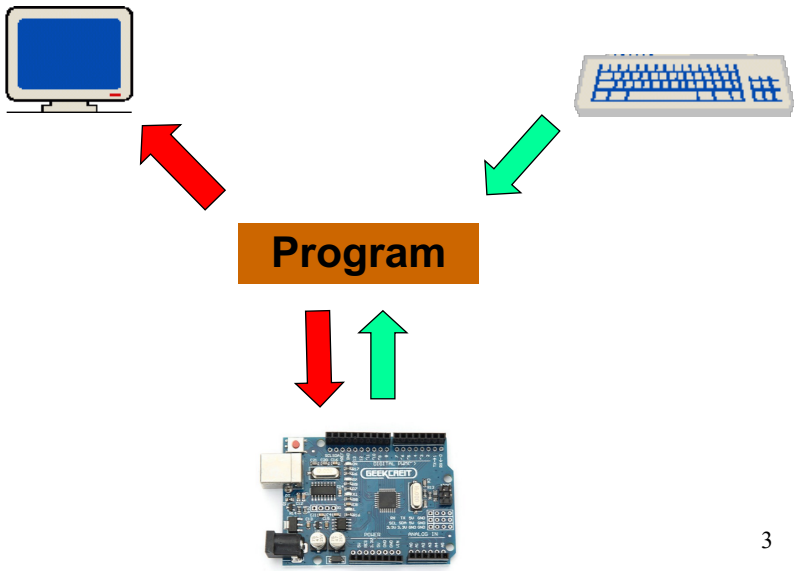
1806ICT
Programming Fundamentals

Input/Output

Topics

- Streams
- Formatted output
- Formatted input

Input/Output



3

3

"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>
```

4

4

`stdout`: Output

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

5

Example: WriteData

Set name to "David"

Set age to 18.2

Set gender to 'M'

Set idNumber to 3825

Output name, age, gender, idNumber

6

```
#include <stdio.h>

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

int main()
{
  char   name[]   = "David" ;
  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;
}
```

7

7

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);
```

8

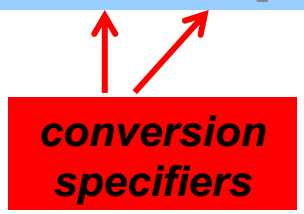
8

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);
```



9

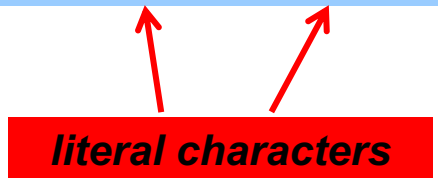
9

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);
```



10

10

printf -- Other-Arguments

- For printf: variables containing data for output

Example:

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

11

printf: Conversion Specifiers

- `c`: display a single character
- `s`: display a character string
- `i` or `d`: display a signed decimal integer
- `u`: display an unsigned decimal integer
- `li` or `ld`: display a signed long integer
- `hi` or `hd`: display a signed short integer
- `f`: display a floating point value (float or double)
- `e` or `E`: display a floating point value in exponential notation
- `Lf`: display long double

12

`stdin:` Input

- Data is read in from **`stdin`** (into a variable) using the **`scanf()`** function

13

13

Example: ReadData

Input name, age, gender, idNumber

14

14

```
#include <stdio.h>
/************************************\
Read in important info about a student
\************************************/
int main()
{
 char name[100] ;        Ashley
 float age ;              19.2
 char gender ;             M
 int idNumber ;          3825

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

Input: Ashley 19.2 M 3825

15

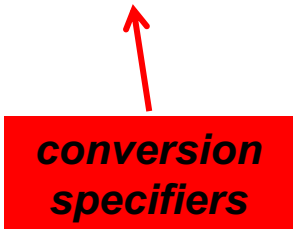
15

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);
```



16

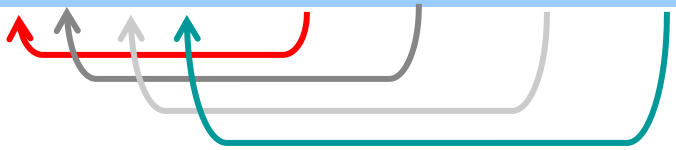
16

scanf -- Other-Arguments

- For scanf: "pointers" to variables where the input will be stored

Example:

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



17

17

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!

18

18

scanf: Conversion Specifiers

- `d`: read an optionally signed decimal integer
- `i`: read an optionally signed decimal, octal, or hexadecimal integer
- `u`: read an unsigned decimal integer
- `c`: read a single character
- `s`: read a sequence of characters
- `f`: read a floating point value

19

19

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

20

20

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);
```

21

21

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);
```

22

22

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);
```

23

Summary

- **Input** from keyboard is via the **stdin** stream
- **Output** to the screen is via the **stdout** stream
- To use the C language I/O functions, you must include the **stdio.h** header file

24

Common Mistakes

- scanf()
  - Forgetting the & in front of the int, float, char variables: program will crash when executed
  - Mismatch between the input value and the conversion specifier: a wrong value is stored in the variable
- printf()
  - Mismatch between the conversion specifier and the variable type: a wrong value is printed out

25

25