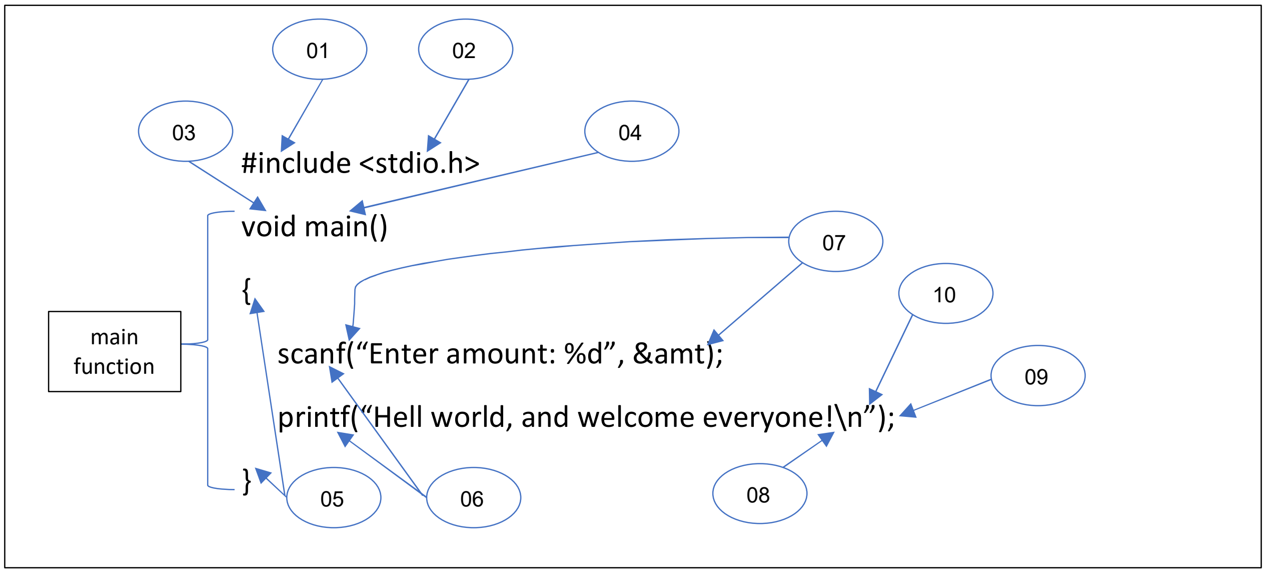
Parts of the C Language Program



```
#include <stdio.h>
void main()
{
    scanf("Enter amount: %d", &amt);
    printf("Hell world, and welcome everyone!\n");
}
```

Figure 1

In all C programs, the starting point is the **main()** function. Every C program has one, even HELLO.C shown above in figure 1. The **main()** function is the engine that makes the program work, which displays the message on the screen.

C programs carry out the tasks in their **main()** function. But whatever is in there, it is the first instruction given to the computer when the program runs.

- **main()** is the name given to the first (or primary) function in every C program. C programs can have other functions, but **main()** is the first one.
- It is a common convention to follow a C language function name with parentheses, as in **main()**. It does not mean anything. Everyone does it, and it is included here so that you do not worry when you see it elsewhere.
- In C Language, you may have seen the error message say, "in function main." This message refers to the main function — the **void main()** part that contains the C language instruction statements you have been writing.
- A function is a procedural process, it is a set of instruction statements that solves the problem's goal. C programs can have many functions in them, though the **main** function is the first function in a C program. It is required.
- The C language is composed of **keywords** that appear in statements. **Statements end in semicolons**, just as sentences in the human's Modern English Language end in periods.

Here are the components in the C program shown in Figure 1:

1. #include is known as a preprocessor directive. What it does is tell the compiler to "include" another *program or file* along with your source code, which generally avoids you writing its source code from scratch.

2. <stdio.h> is a filename embraced by angle brackets. The whole statement #include <stdio.h> tells the compiler to use the file STDIO.H, which contains standard input/output, commands required by C programs. Functions such as printf for display output, and scanf for keyboard input.

3. void main identifies the name of the function main. The void identifies the type of function or what the function produces. In the case of main, it does not return anything, and the C term for that is "void."

4. Two empty **parentheses** follow the function name. Sometimes, there may be items in these parentheses. E.g. passing of argument(s) values.

5. The **curly brackets or braces** enclose the function, embracing in all its parts. Everything between { and } is part of the function main() in Figure 1.

6. scanf and printf are C language instruction statements, part of the programming language that eventually tells the computer to execute.

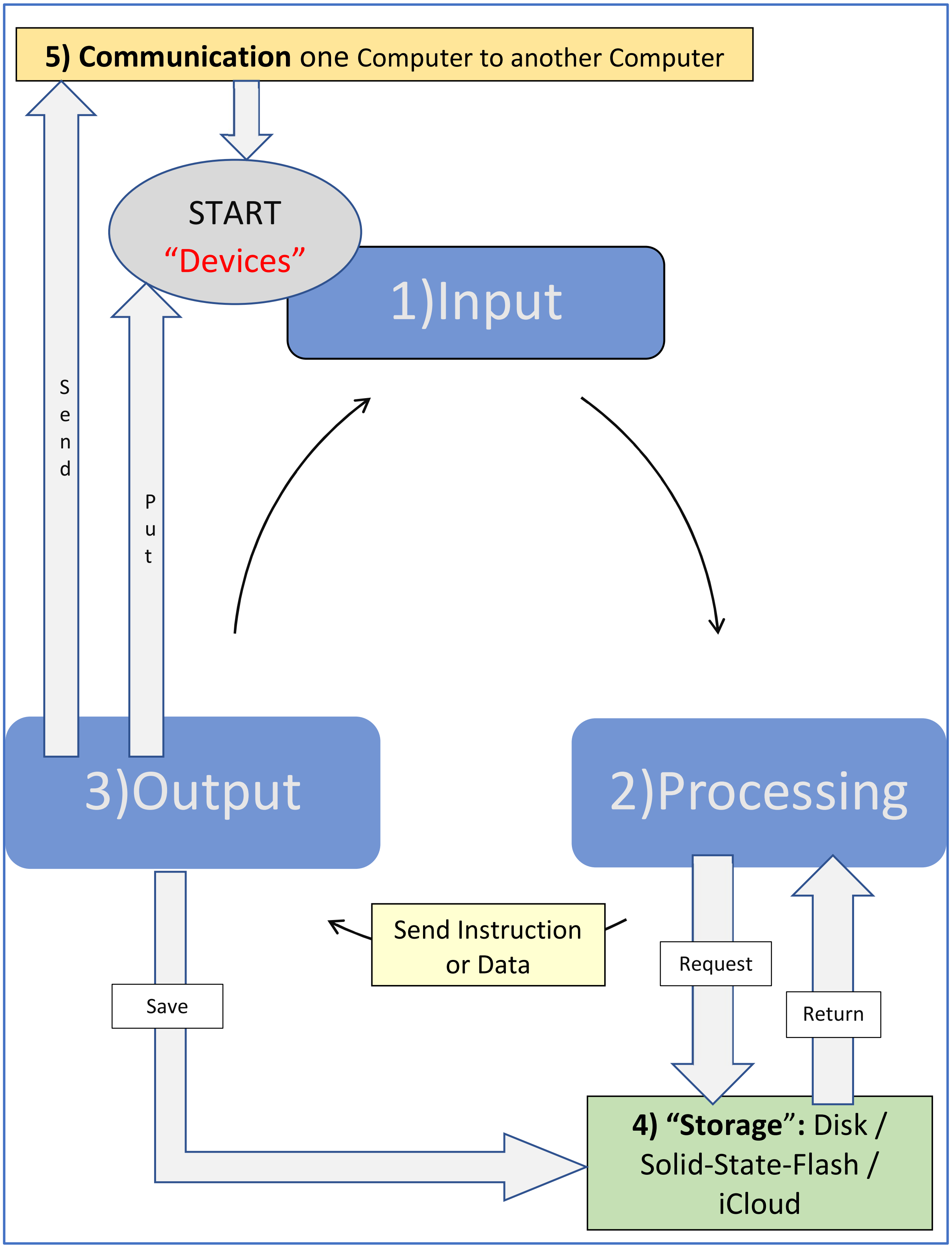
7. Belonging to scanf and printf are more parentheses. In this case, the parentheses enclose text, or a **"string literal" of text**. Everything between the **double quotes** (") is part of scanf's or printf's text string.

8. An interesting part of the text string is **\n**. That is the backslash character and a little n. What it represents is the character produced by pressing the keyboard Enter key. What it does is to end the text string with a **"new line."**

9. Finally, the scanf and printf line, or statement, ends with a semicolon. The semicolon is how the C compiler knows when one statement ends and another begins. *The semicolon is still required even if it is the only instruction in the program.*

10. Text in a program is referred to as a *string.* For example, "hello" is a string of text. The string is enclosed by double quotes.

The five major functions of a computer system: **Input, Processing, Output, Storage (optional), and Communication (computer to computer)**.


5) Communication one Computer to another Computer
START
"Devices"
1)Input
S e n d
P u t
3)Output
2)Processing
Send Instruction or Data
Save
Request
Return
4) "Storage": Disk / Solid-State-Flash / iCloud