Lab # 3: I/O Programming

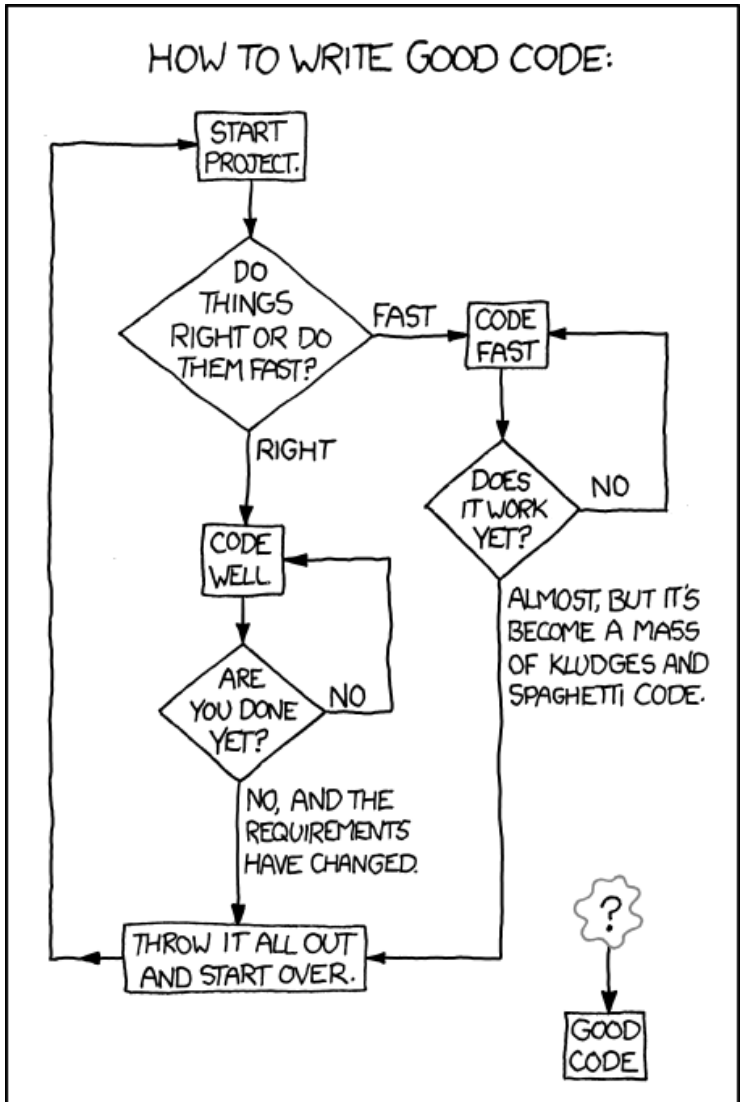
EC-102 – Computer Systems and Programming

Usama Wajhi

School of Mechanical and Manufacturing Engineering (SMME),
National University of Sciences and Technology (NUST)

October 8, 2016

Outline



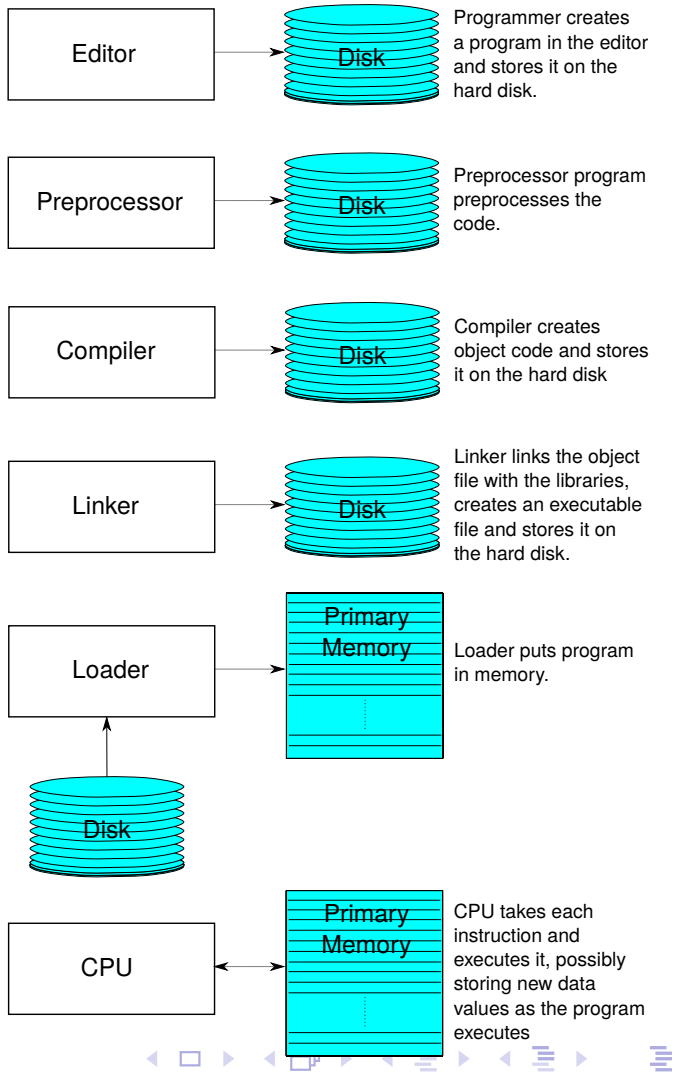
Quiz

1. Write short description of the following: (Marks:2 for each part)
   - Algorithm
   - Machine Language
   - Assembler
   - Advantages of Debugging?
2. Advantages of C++ in detail (Marks:5)
3. Applications of Programming (Marks:2)

Basics of a Typical C++ Environment

Phases of C++ Programs

- Edit
- Preprocess
- Compile
- Link
- Load
- Execute



First Program in C++

```
1 // my first program in C++
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     cout << "My First Program!
    " << endl;
7     return 0;
8 }
```

Line-by-line Explanation

1. Single line comment. In C++, there are two types of comments.
   - Single line – //...
   - Multi line – /*...*/
2. Preprocessor directive to include input/output stream header file.
3. A namespace where features of the C++ standard library such as `cout` or `endl` are declared.

First Program in C++

```
1 // my first program in C++
2 #include <iostream>
3 using namespace std;
4 int main()
5 {
6     cout << "My First Program!
    " << endl;
7     return 0;
8 }
```

Line-by-line Explanation

4. Function `main` appears at least once in every C++ program.
5. Left brace begins function body.
6. cout belongs to namespace std and is used for console output, << is known as a stream insertion operator. `endl` also belongs to namespace std and is used to insert a newline

Arithmetic Operators

C++ operation	C++ arithmetic operator
Addition	+
Subtraction	-
Multiplication	*
Division	/
Modulus	%

Fundamental C++ Data Types

The C++ language offers several fundamental data types. These include

- `int` (-2,147,483,648 *to* 2,147,483,647) takes up **4 bytes**
- `short` (-32,768 *to* 32,767) takes up **2 bytes**
- `float` (3.4×10^{-38} *to* 3.4×10^{38}) takes up **4 bytes**
- `double` (1.7×10^{-308} *to* 1.7×10^{308}) takes up **8 bytes**
- `char` (-128 *to* 127) takes up **1 byte**

Fundamental C++ Data Types

- The strong data type system of C++ helps make sure that the data variables are used consistently and correctly.
- Type checking makes it easy for the compiler to spot errors during compilation and thus prevent such issues during execution of the program.
- Before a variable is used in C++, it must be declared and defined as follows,
  `int myage;`
- This line declares and defines a variable named `myage` as an integer.

Fundamental C++ Data Types

- A declaration introduces the name `myage` to the compiler and attaches a specific meaning to it.
- A definition like this also instructs the compiler to allocate some memory for the variable.

Fundamental C++ Data Types

When the compiler reads `myage` definition,

- It sets aside enough memory storage for an integer and uses the name `myage` to refer to it.
- It reserves the name `myage` so that it cannot be used by any other variable.
- It ensures that whenever this variable is used, it is used in a way that is consistent with the way an integer should be used.

Keywords in C++

- Predefined reserved identifiers which have a special significance within the language.
- They are case-sensitive and cannot be used as identifiers in your program.
- They will be highlighted with a specific color by Visual Studio's editor as you write your code.
- If the keywords you type in do not appear highlighted, then they have been entered incorrectly.

Keywords in C++

Here's a list of all the reserved keywords in standard C++.

alignas	continue	friend	register	true
alignof	decltype	goto	reinterpret_cast	try
asm	default	if	return	typedef
auto	delete	inline	short	typeid
bool	do	int	signed	typename
break	double	long	sizeof	union
case	dynamic_cast	mutable	static	unsigned
catch	else	namespace	static_assert	using
char	enum	new	static_cast	virtual
char16_t	explicit	noexcept	struct	void
char32_t	export	nullptr	switch	volatile
class	extern	operator	template	wchar_t
const	false	private	this	while
constexpr	float	protected	thread_local	
const_cast	for	public	throw	

Stream I/O

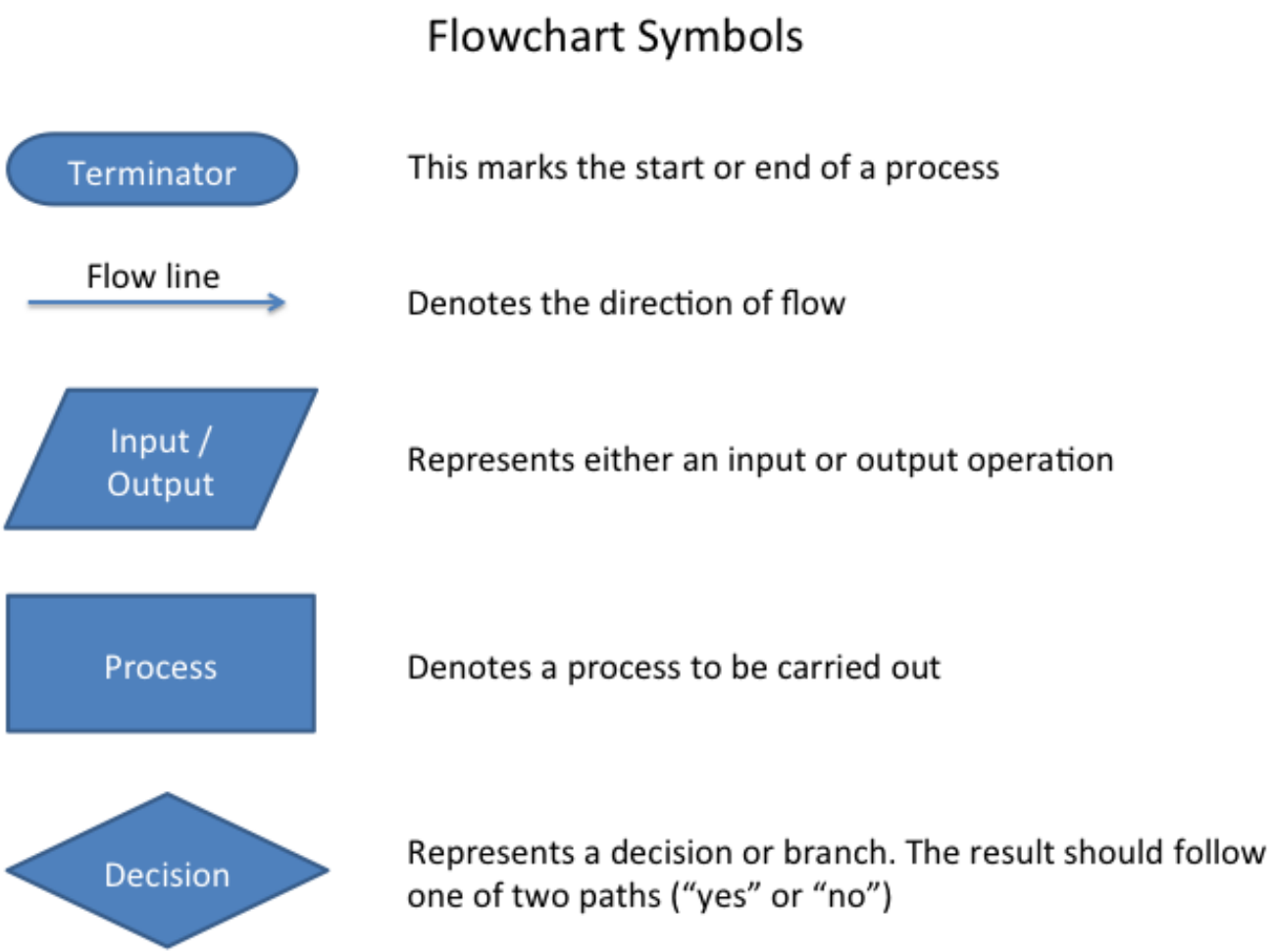
- C++ input/output revolves around the notion of a data stream, where we can insert data into an output stream or extract data from an input stream.
- The standard output stream to the screen is referred to as `cout`.
- The standard input stream from the keyboard is referred to as `cin`.

Assignment Statements

- The statement which assigns some value to a variable is called an assignment statement.
- The $=$ operator is used to assign a value to a variable.
- In an assignment statement such as `myage = 25;`, the variable `myage` has been assigned a value of 25.

What is Flowchart?

Graphical representation of Code.

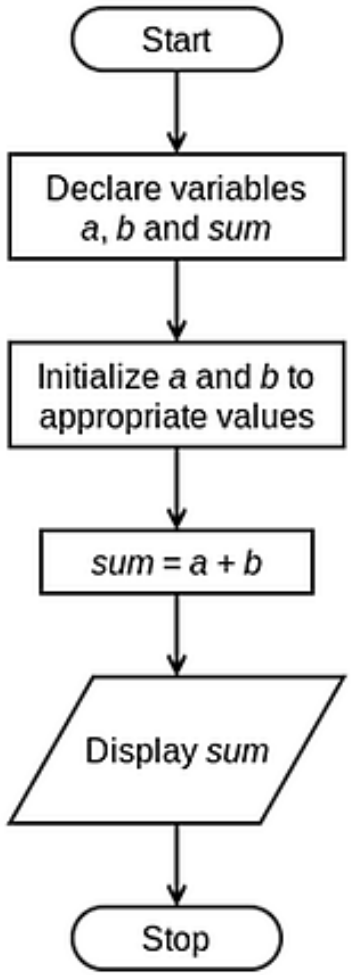


Example 1

Algorithm

1. Start
2. Declare variables `a`, `b` and `sum`
3. Initialize `a` and `b` to appropriate values
4. Add `a` and `b` and assign the result to `sum`
5. Display `sum`
6. Stop

Flowchart

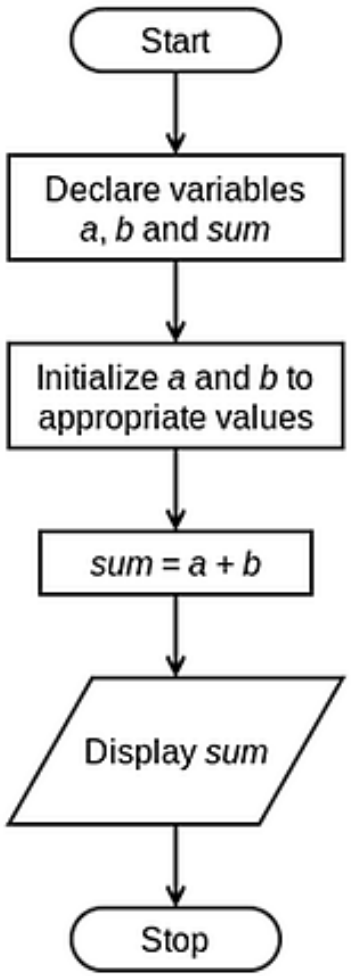


Example 1

Code

```
1  // this program displays the
       sum of two numbers
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int a, b, sum;
7      a = 20;
8      b = 30;
9
10     sum = a + b;
11
12     cout << sum << endl;
13     return 0;
14 }
```

Flowchart

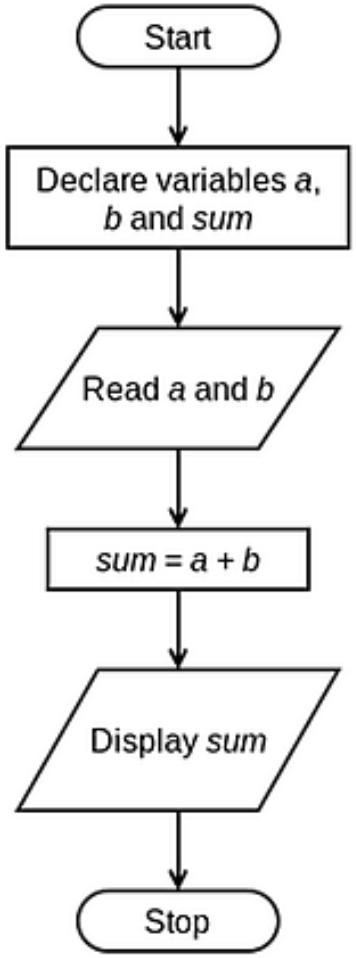


Example 2

Algorithm

1. Start
2. Declare variables `a`, `b` and `sum`
3. Read values of `a` and `b`
4. Add `a` and `b` and assign the result to `sum`
5. Display `sum`
6. Stop

Flowchart



Example 2

Code

```
1  // this program displays the sum of
       two numbers entered by the user
2  #include <iostream>
3  using namespace std;
4  int main()
5  {
6      int a, b, sum;
7      cout << "Enter first number: ";
8      cin >> a;
9      cout << "Enter second number: ";
10     cin >> b;
11
12     sum = a + b;
13
14     cout << "The sum of two numbers
       is: ";
15     cout << sum << endl;
16
```

Flowchart