|  |
| --- |
| CS-111 |

**Fundamentals of Computer**

|  |
| --- |
| **Programming** |

|  |
| --- |
| Lecture 2 (Part – I) |

|  |
| --- |
| Program Development |

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | **Course Instructor: Qurrat-ul-ain Babar** | **Institute of Geographical Information Systems NUST Institute of Civil**  **Engineering (NICE)** | |

|  |
| --- |
|  |

**Road Map for Today**

• RECAP  
• Developing a program  
• Introduction to C++  
• Typical C++ Environment

|  |
| --- |
|  |

|  |
| --- |
|  |

**Developing a Program**

|  |
| --- |
|  |

**Problem Solving and Implementation**

A programming task can be divided into two phases:

**1. *Problem Solving***  
– Define: Clearly describe a problem  
– Design its solution: Produce an ordered   
 sequence of steps that describes solution to the problem

**2. *Implementation Phase***  
– Implement the program in some programming language  
– Write code, compile, link, test, debug

4

**Problem Solving and Implementation**

|  |  |
| --- | --- |
| 5 |  |

**Define a Problem:**

Break the definition of the problem down into manageable steps. Example: input, processing, output

**Display whether the temperature entered by the**  **user is below or above freezing point?**

**Input; Read the temperature from keyboard**

**Processing; Test the temperature – below or above**  **freezing point**

**Output; Display the result on screen**

6

**Design the solution**

**Algorithm:**  
• A precise rule or set of rules specifying how to solve a problem

• A sequence of language independent steps which may be followed to solve a problem.

An algorithm can be developed with a:

**Pseudo code**

**Flowchart**

7

**Algorithm**

**Pseudo code**  
• Artificial, informal language that helps us develop algorithms  
• Similar to everyday English  
• Not actually executed on Computers  
• Helps us think out a program before   
 writing it

Easy to convert into a corresponding C++ program

Consists only of executable statements

8

**Algorithm**

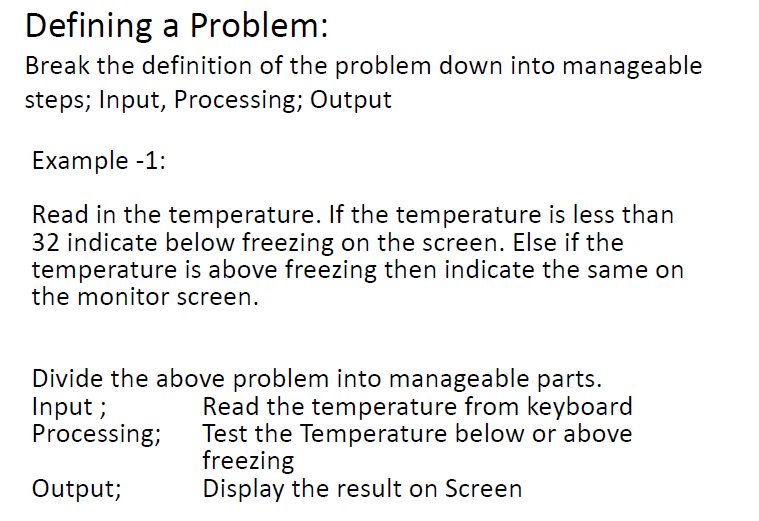
**Flowchart**  
• Diagrammatic or graphical represenations of steps for solving the given problem.

• Use standard symbols developed by ANSI (American National Standard Institute)

9

**Building Blocks of Flowchart** Start and End

|  |  |  |
| --- | --- | --- |
|  | | |
| 10 | Connector |  |
| Arrows |  |
| Processing Steps | |  | | --- | |  | |
| Input/ Output |  |
| Decision |  |

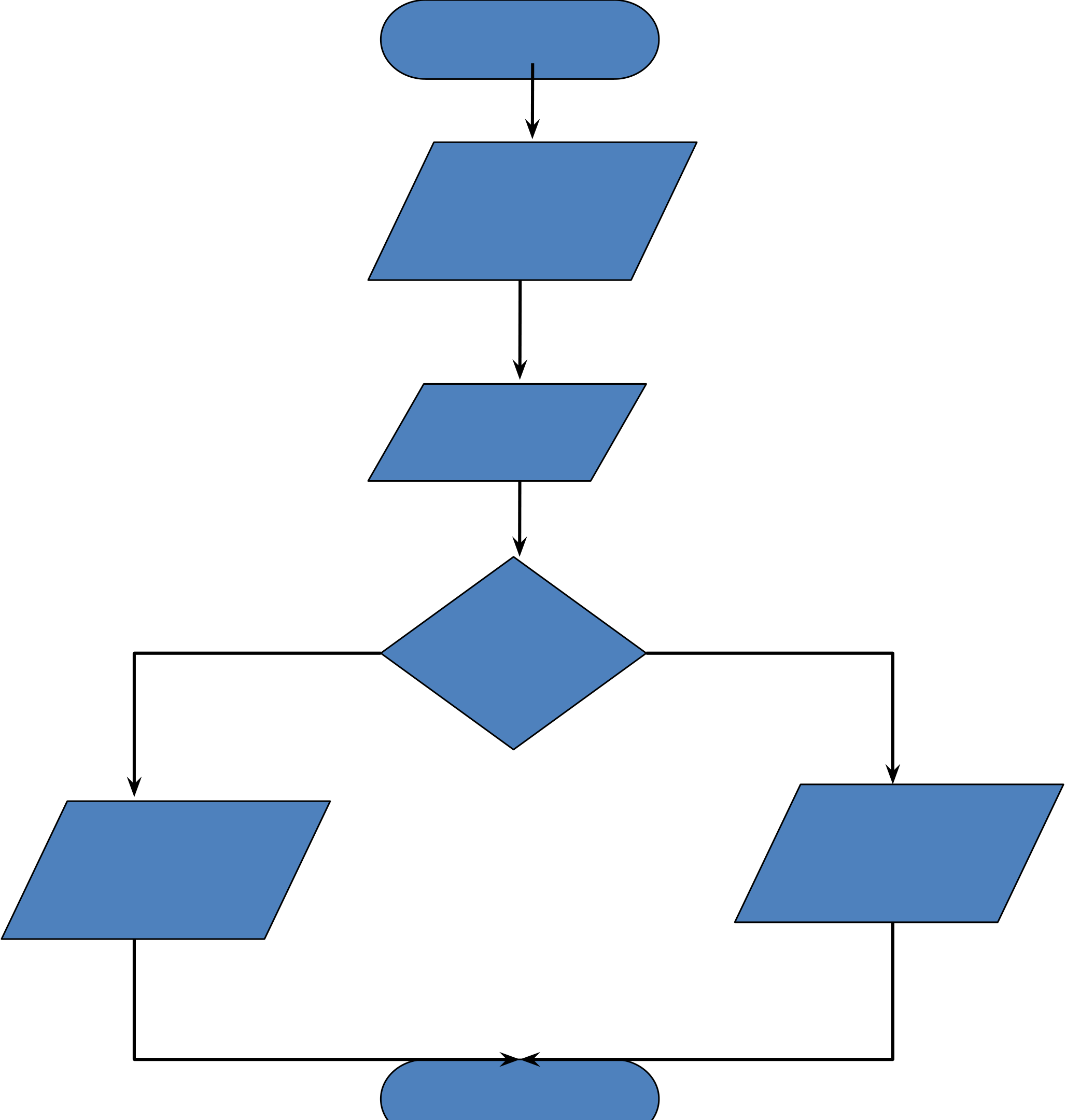
**0**

11

**Algorithm – Pseudo code**

Example -1:   
Print “Enter Temperature”  
Read the Temp   
if (Temp < 0) then   
Print “BELOW FREEZING”  
else   
Print “ABOVE FREEZING”  
endif

12



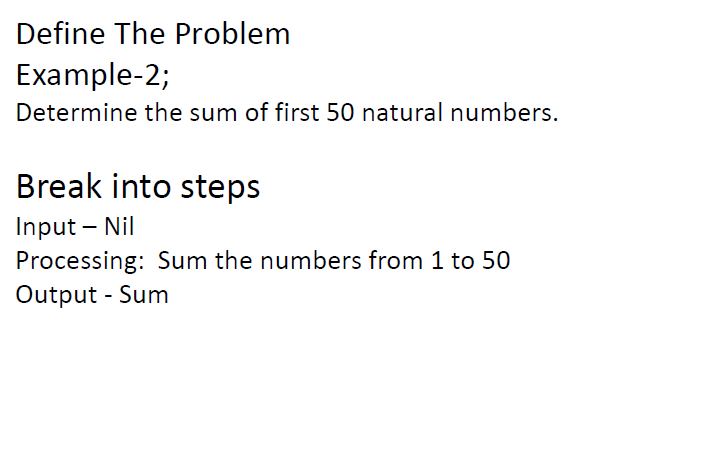
|  |  |
| --- | --- |
| **Algorithm –Flowchart** | **Start**  **Print** **“Enter**  **Temp”** |

**Read**

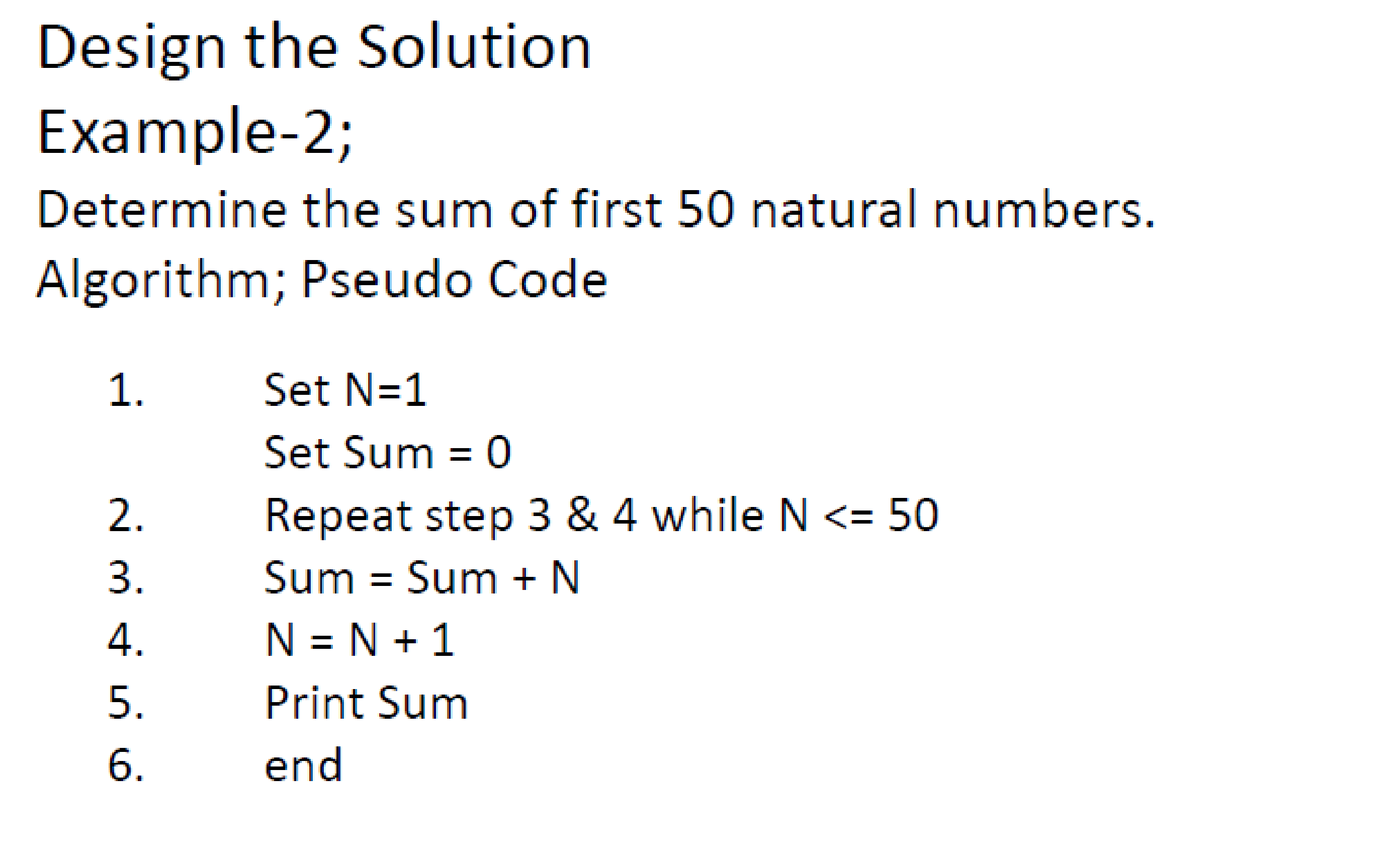
**Temp**

|  |  |  |
| --- | --- | --- |
| **Yes** | **Temp** | **No** |
| **<0?** |

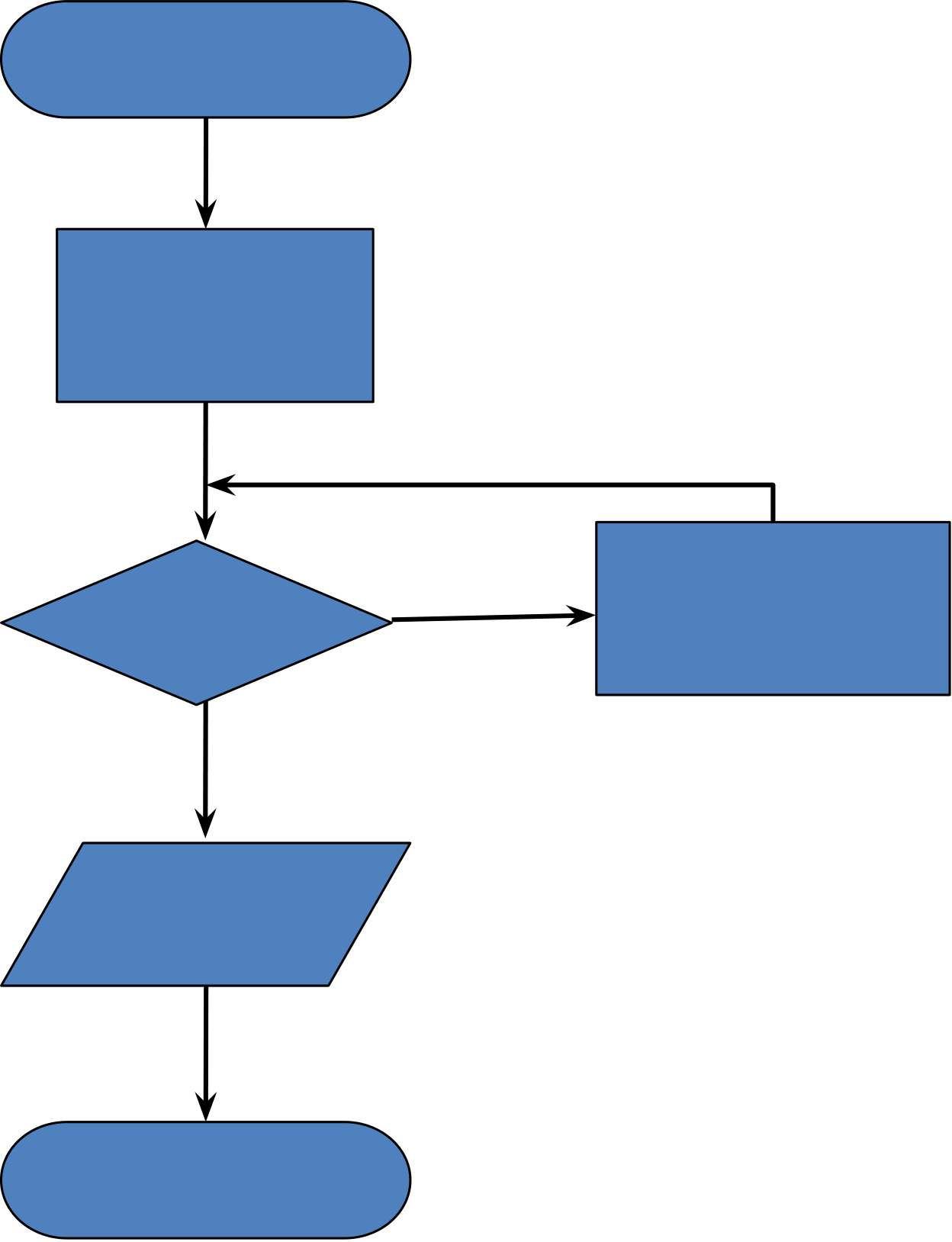
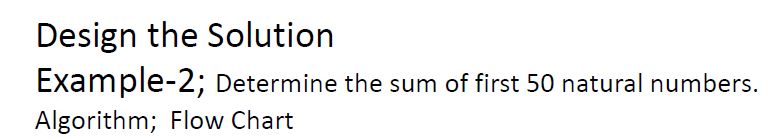
|  |  |  |  |
| --- | --- | --- | --- |
| 13 | **Print** | **End** | **Print** |
| **“Above** |
| **“Below** |
| **Freezing”** |
| **Freezing”** |



14



15

**Start**

**Sum=0**

**N=1**

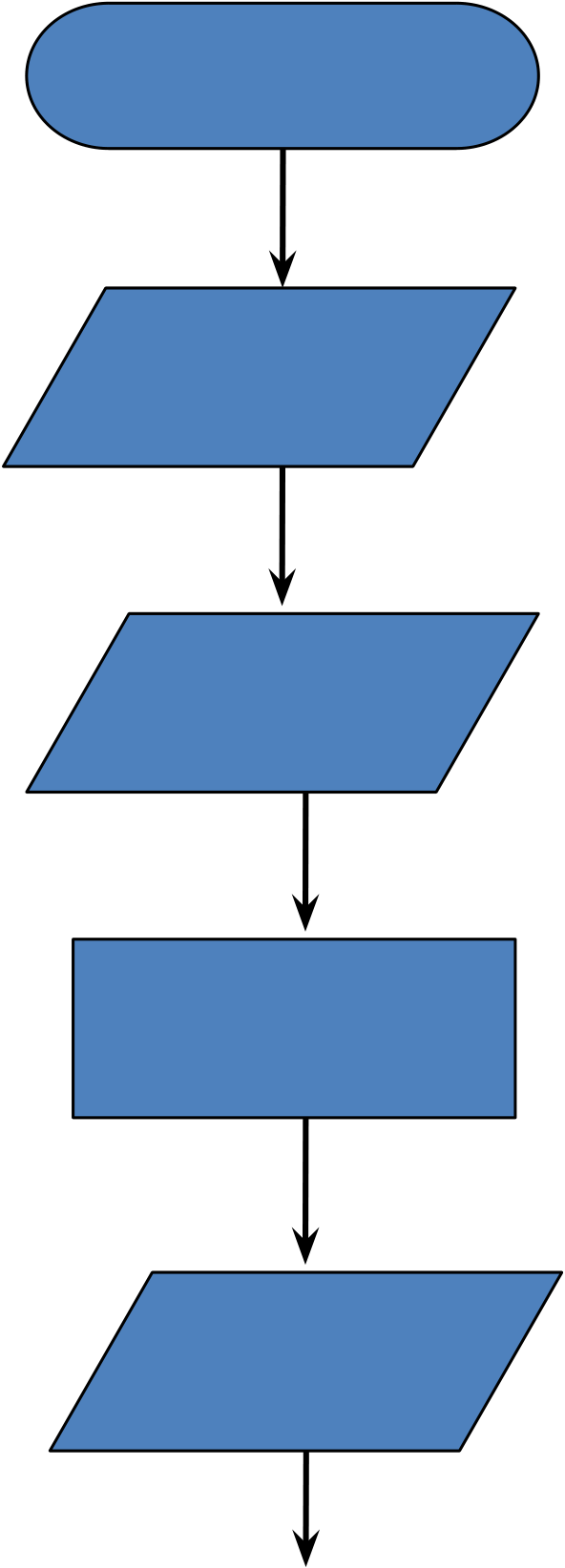
|  |  |  |
| --- | --- | --- |
| **N=51** | **No** | **Sum=Sum+** |
| **N** |
| **Yes** | **N=N+1** |

**Print**

**Sum**

**End**

16

**Example 3: Adding two Numbers**  **Start**

|  |  |
| --- | --- |
|  | **Read**  **num1**  **Read**  **num2**  **Sum=num1+ num2**  **Print**  **Sum** |
| 17 | **End** |

|  |
| --- |
|  |

**Introduction to C++**

|  |
| --- |
|  |

Introduction to C++

• Where did C++ come from?

– Derived from the C language  
– C was derived from the B language  
– It was originally called **“C with Classes”** and was renamed C++ in 1983

• Why the ‘++’?

– ++ is shorthand for adding 1 to a number in programming, so C++ roughly means  
**“one better than C.”**

Copyright © 2003 Pearson Education,   
Inc.

C++ History

• C developed by Dennis Ritchie at AT&T Bell Labs in the 1970s.

– Used to maintain UNIX systems  
– Many commercial applications written in c

• C++ developed by Bjarne Stroustrup at AT&T Bell Labs in the 1980s.

– Overcame several shortcomings of C  
– Incorporated object oriented programming– C remains a subset of C++

Copyright © 2003 Pearson Education,   
Inc.

A Sample C++ Program• A simple C++ program begins this way   
 **#include <iostream>**   
 **using namespace std;**   
 **int main()**   
 **{**  
• And ends this way

|  |  |
| --- | --- |
| **}** | **return 0;** |

Copyright © 2003 Pearson Education,

Inc.

|  |
| --- |
|  |

Typical C++ Environment

|  |
| --- |
|  |



Basics of a Typical C++ Environment

• C++ systems

– Program-development environment

– Language

– C++ Standard Library

• C++ program names extensions

– .cpp

– .cxx

– .cc

– .C

© 2003 Prentice Hall, Inc. All rights reserved.

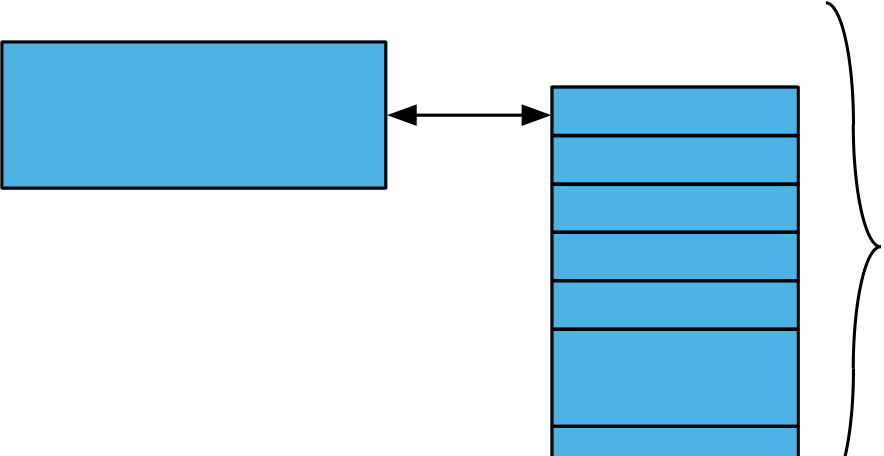
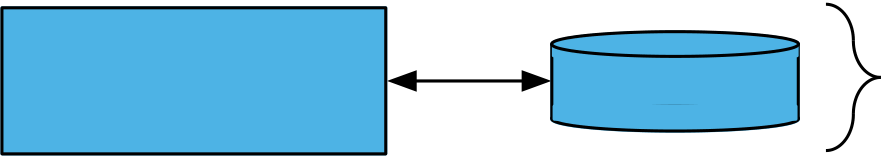
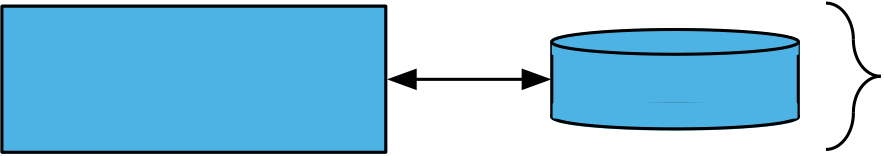
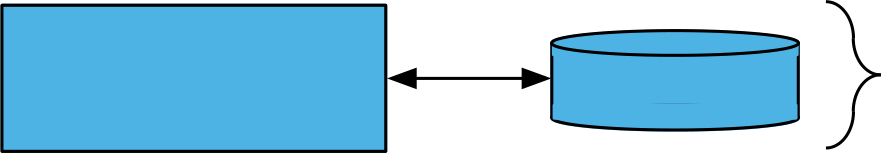
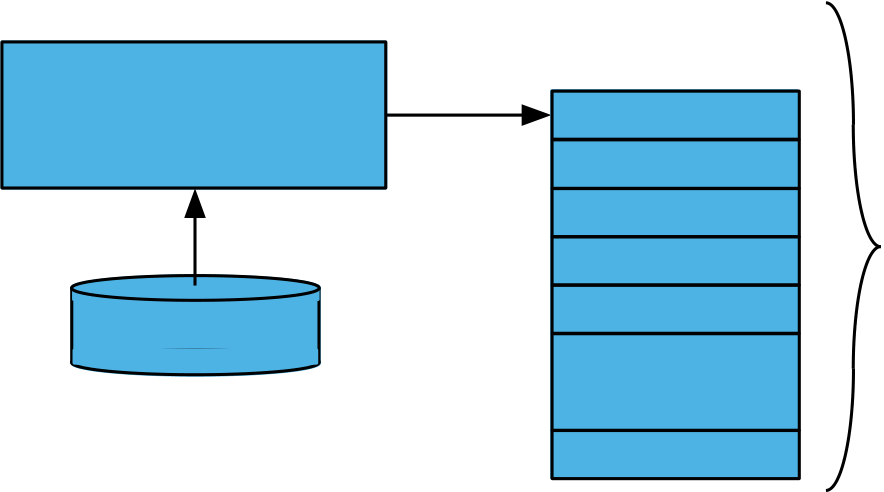
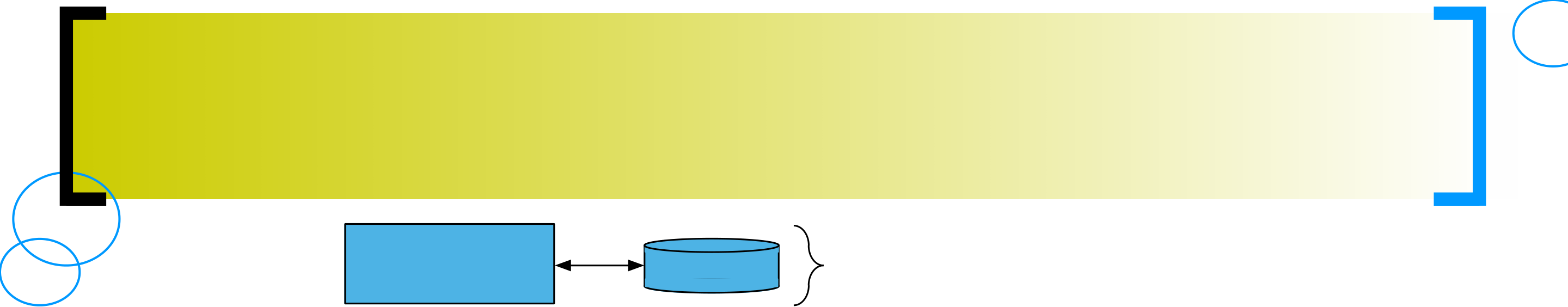


24 Basics of a Typical C++ Environment

Phases of C++ Programs:

1. Edit   
2. Preprocess   
3. Compile   
4. Link   
5. Load   
6. Execute

© 2003 Prentice Hall, Inc. All rights reserved.



25

Basics of a Typical C++ Environment

|  |  |  |
| --- | --- | --- |
| Editor | Disk | Program is created in the editor and |
| stored on disk. |
| Preprocessor | Disk | Preprocessor program processes the |
| Compiler | code. |
| Disk | Compiler creates object code |
| and stores it on disk. |
| Linker | Disk | Linker links the object code with the |
| libraries, Creates an executable file |
| Loader | Primary | and stores it on disk |
| Memory |

Loader puts program

|  |  |  |
| --- | --- | --- |
| Disk | **. .** | in memory. |

**. .**

**. .**

Primary

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CPU | Memory | CPU | takes | | each | instruction | | and |
| executes | | it, | possibly | | storing | new |

data values as the program executes.

**. .**

|  |  |
| --- | --- |
| © 2003 Prentice Hall, Inc. All rights reserved. | **. .**  **. .** |



26

Basics of a Typical C++ Environment

• Common Input/output functions

– cin (See-in)

•Standard input stream

•Normally keyboard

– cout (See-out)

•Standard output stream

•Normally computer screen

– cerr (See-err)

•Standard error stream

•Display error messages

© 2003 Prentice Hall, Inc. All rights reserved.

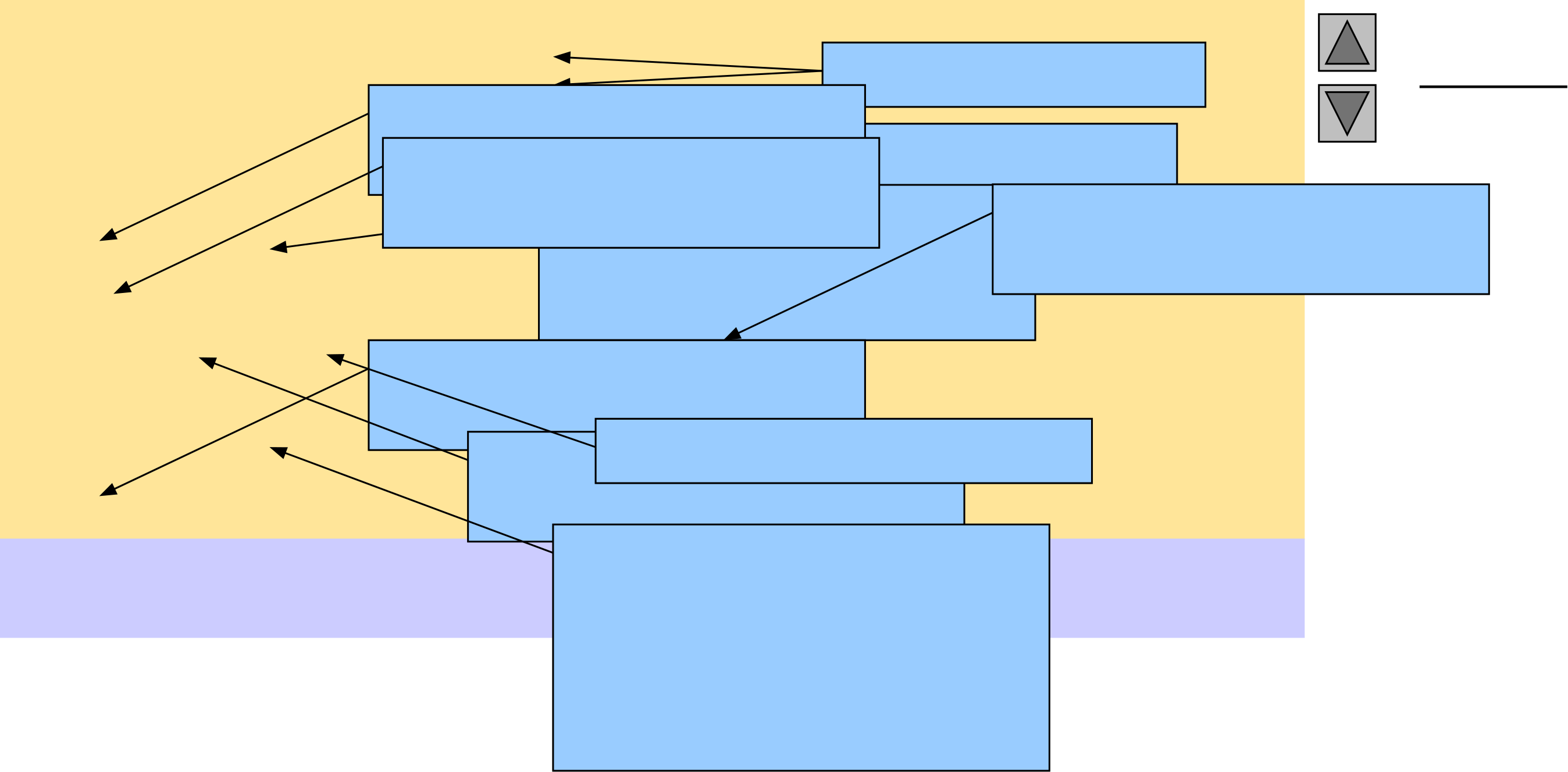


27

A Simple Program: Printing a Line of Text

• Before writing the programs  
 **–** **Comments**  
 •Document programs  
 •Improve program readability  
 •  
Ignored by compiler •Single-line comment  
 –Begin with //  
 •Multiple-line comment  
 –  
Begin with /\* and end with \*/ **–/\* … \*/**  
 **–** **Preprocessor directives**  
 •Processed by preprocessor before compiling •Begin with #

© 2003 Prentice Hall, Inc. All rights reserved.



|  |  |  |  |
| --- | --- | --- | --- |
| 1 // Fig. 1.2: fig01\_02.cpp  2 // A first program in C++. Single-line Outline  3 #include <iostream> Function main returns comments.  4  5 // function main begins program execution  6 int main() an integer value. Left brace { begins function body. Function main appears exactly once in every Preprocessor directive to include input/output stream header file Statements end with a semicolon ;. fig01\_02.cpp (1 of 1)  7 {  8 std::cout << "Welcome to C++!\n"; C++ program.. <iostream>. fig01\_02.cpp | | | 28 |
| 9 Corresponding right  10 return0; // indicate that program ended successfully brace } ends function  11  12 } // end function main body. Name cout belongs to namespace std. Stream insertion operator. | | output (1 of 1) |
| Welcome to C++! | Keyword return is one  of several means to exit | |

function; value 0   
indicates program   
terminated successfully.

© 2003 Prentice Hall,   
Inc.

All rights reserved