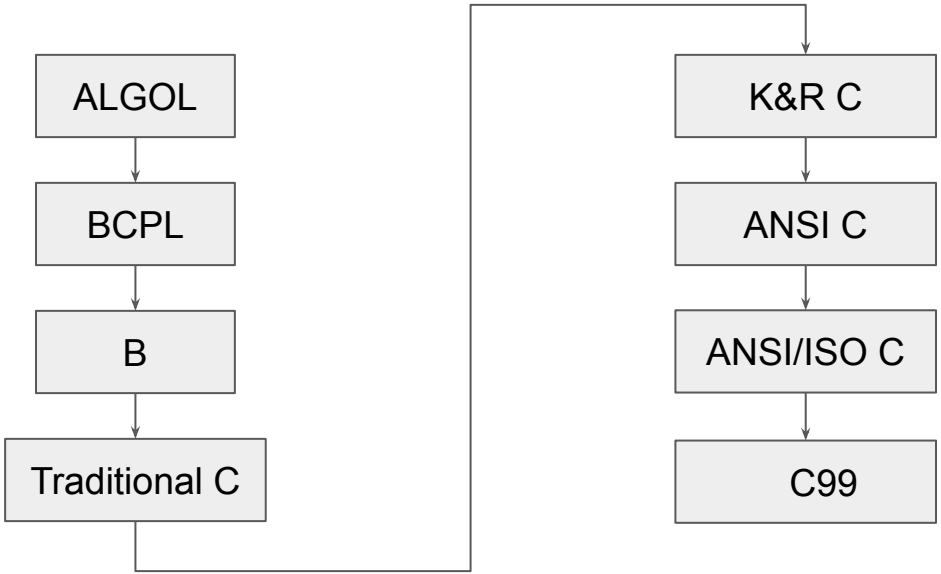
Overview of Programming

Dr. Nachiket Tapas

History of C

C is one of the most popular computer languages today because it is a structured, high-level, machine independent language.



Evolution

- The root of all modern languages is ALGOL, introduced in the early 1960s.
- In 1967, Martin Richards developed a language called BCPL (Basic Combined Programming Language) primarily for writing system software.
- In 1970, Ken Thompson created a language using many features of BCPL and called it simply B.
- Traditional C was evolved from ALGOL, BCPL, and B by Dennis Ritchie at the Bell Laboratories in 1972.
- Brian Kerningham and Dennis Ritchie made C popular and was called K&R C.
- American National Standard Institute (ANSI) defined standard for C 1989.
- Approved by International Standards Organization (ISO) in 1990.
- C99 = C + certain features of C++ and Java.

Why program in a high-level language like C?

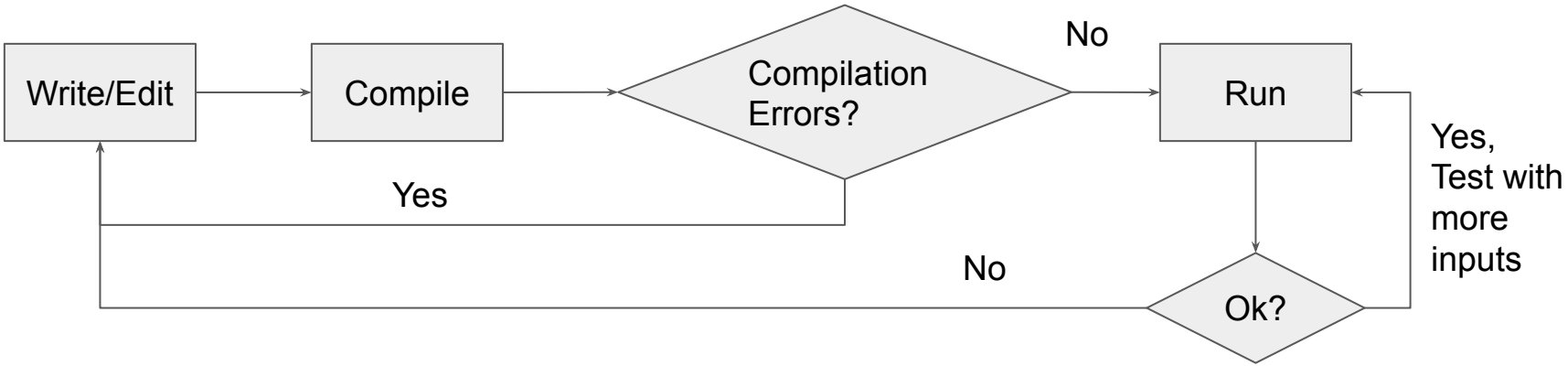
```
b8    21 0a 00 00   #moving "!\n" into eax
a3    0c 10 00 06   #moving eax into first memory location
b8    6f 72 6c 64   #moving "orld" into eax
a3    08 10 00 06   #moving eax into next memory location
b8    6f 2c 20 57   #moving "o, W" into eax
a3    04 10 00 06   #moving eax into next memory location
b8    48 65 6c 6c   #moving "Hell" into eax
a3    00 10 00 06   #moving eax into next memory location
b9    00 10 00 06   #moving pointer to start of memory location into ecx
ba    10 00 00 00   #moving string size into edx
bb    01 00 00 00   #moving "stdout" number to ebx
b8    04 00 00 00   #moving "print out" syscall number to eax
cd    80            #calling the linux kernel to execute our print to stdout
b8    01 00 00 00   #moving "sys exit" call number to eax
cd    80            #executing it via linux sys_call
```

Why program in a high-level language like C?

- Writing programs in machine language is long, tedious, and error-prone.
- They are not portable - meaning program written for one machine may not work on another machine.
- Compilers work as a bridge
- Take as input a C program and produce an equivalent machine program.

The Programming Cycle

1. Write you program or edit (i.e. change or modify) your program.
2. Compile your program. If compilation fails, return to editing step.
3. Run your program on an input. If output is not correct, return to editing step.
    a. Repeat step 3 for other inputs if any.



Simple Program

```
#include<stdio.h>
void main() {
        // First program
        printf(“Welcome to CSVTU”);
}
```

The program prints the message:

Welcome to CSVTU

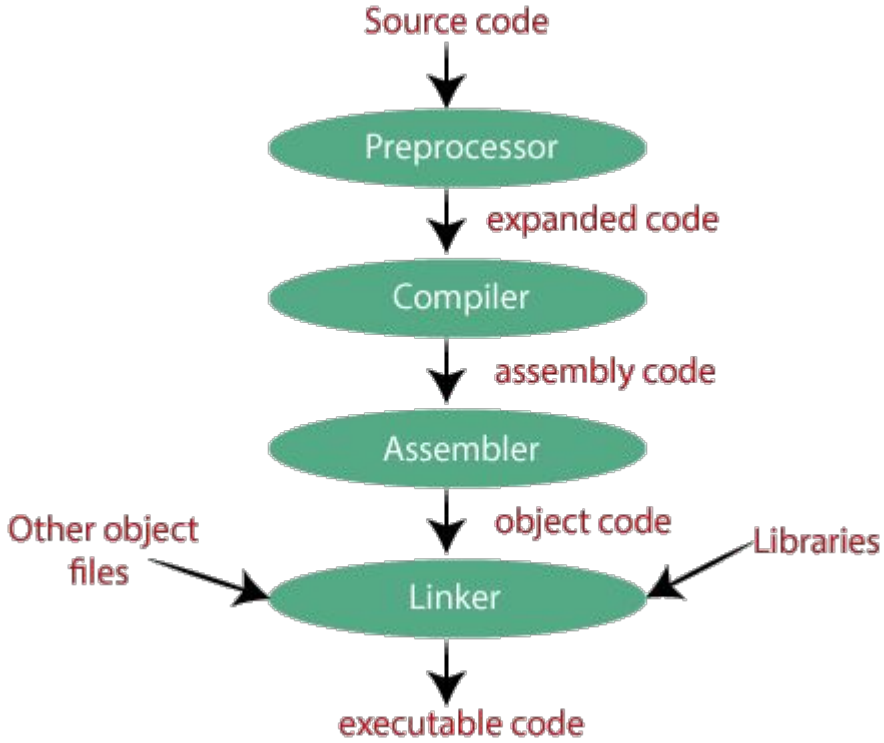
Program Compilation Process

Preprocessor

Compiler

Assembler

Linker



Preprocessor

The preprocessor has several roles:

- it gets rid of all the comments in the source file(s)
- it includes the code of the header file(s), which is a file with extension .h which contains C function declarations and macro definitions
- it replaces all of the macros (fragments of code which have been given a name) by their values

The output of this step will be stored in a file with a “.i” extension, so here it will be in main.i.

Compiler

The compiler will take the preprocessed file and generate IR code (Intermediate Representation), so this will produce a “.s” file.

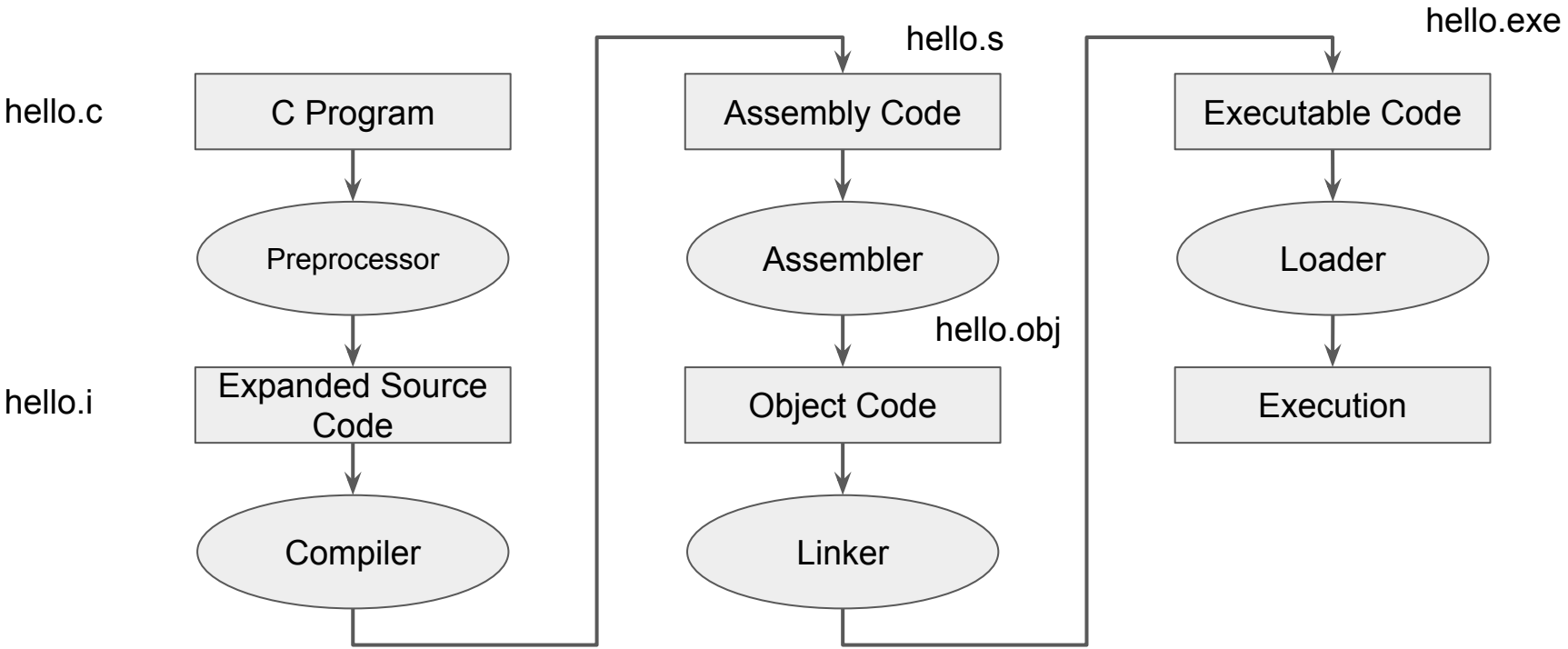
Assembler

The assembler takes the IR code and transforms it into object code, that is code in machine language (i.e. binary). This will produce a file ending in “.o”.

Linker

- The linker creates the final executable, in binary.
  - linking all the source files together, that is all the other object codes in the project.
  - linking function calls with their definitions.

Intermediate Files



Header File

#include<stdio.h>

- This tells C compiler to include standard input output library.

Main function

- In C, a “name” followed by “()” is called a function.
- main() is a special function used by C language to tell the computer where the programs start.
- Every program must have exactly one main() function.
- Curly brace - “{” shows the beginning of function main.
- Curly brace - “}” shows the ending of function main.
- Everything between “{” and “}” is part of main function.

Comments

- Anything starting with “//” or “/*” is known as comment.
- Comments are used to make the program readable.
- Comments are not executed by the compiler.

printf function

- printf is a predefined standard C function for printing output.
- Anything between () are called parameters. For example printf(“ABCD”).
- Predefined means that the code for printf is already written and can be used as is.
- printf causes everything written in between “” to be printed as it is. For example, we printed Welcome to CSVTU

Program to add two numbers

```
/* Program to add two numbers */
main() {
        int number;
        float amount;
        number = 100;
        amount = 75.35 + 30.75;
        printf(“%d\n”, number);
        printf(“%5.2f”, amount);
}
```

What will be the output?

100

106.10

Programs to try

Write a program to print the following:

First Name: Your First Name

Last Name: Your Last Name

University: CSVTU

Example:

First Name: Nachiket

Last Name: Tapas

University: CSVTU

Programs to try

Program to add two integers 20 and 50.

Program to add three floating point numbers 10.5, 11.5, and 12.5.

Thank You!!