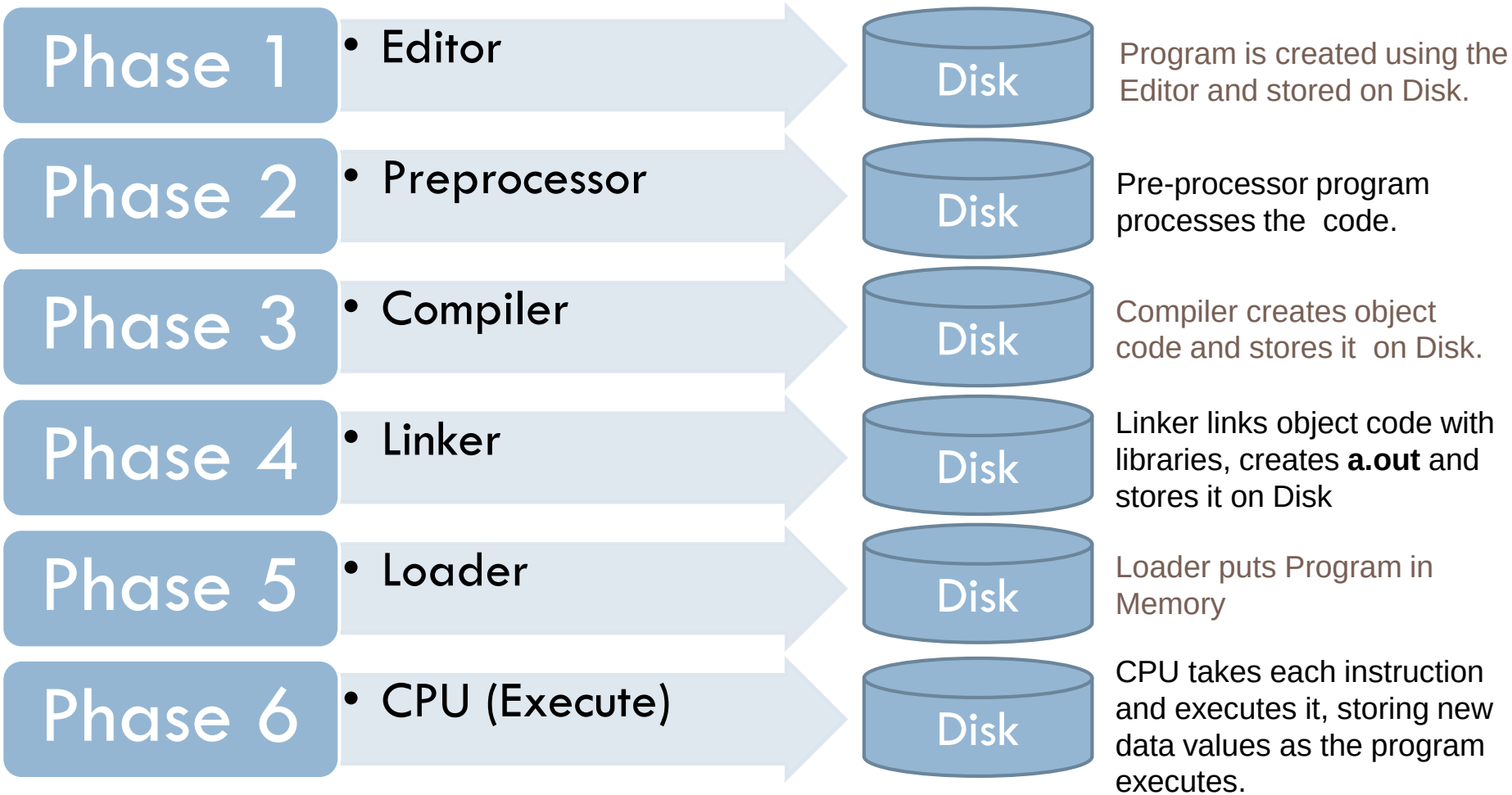
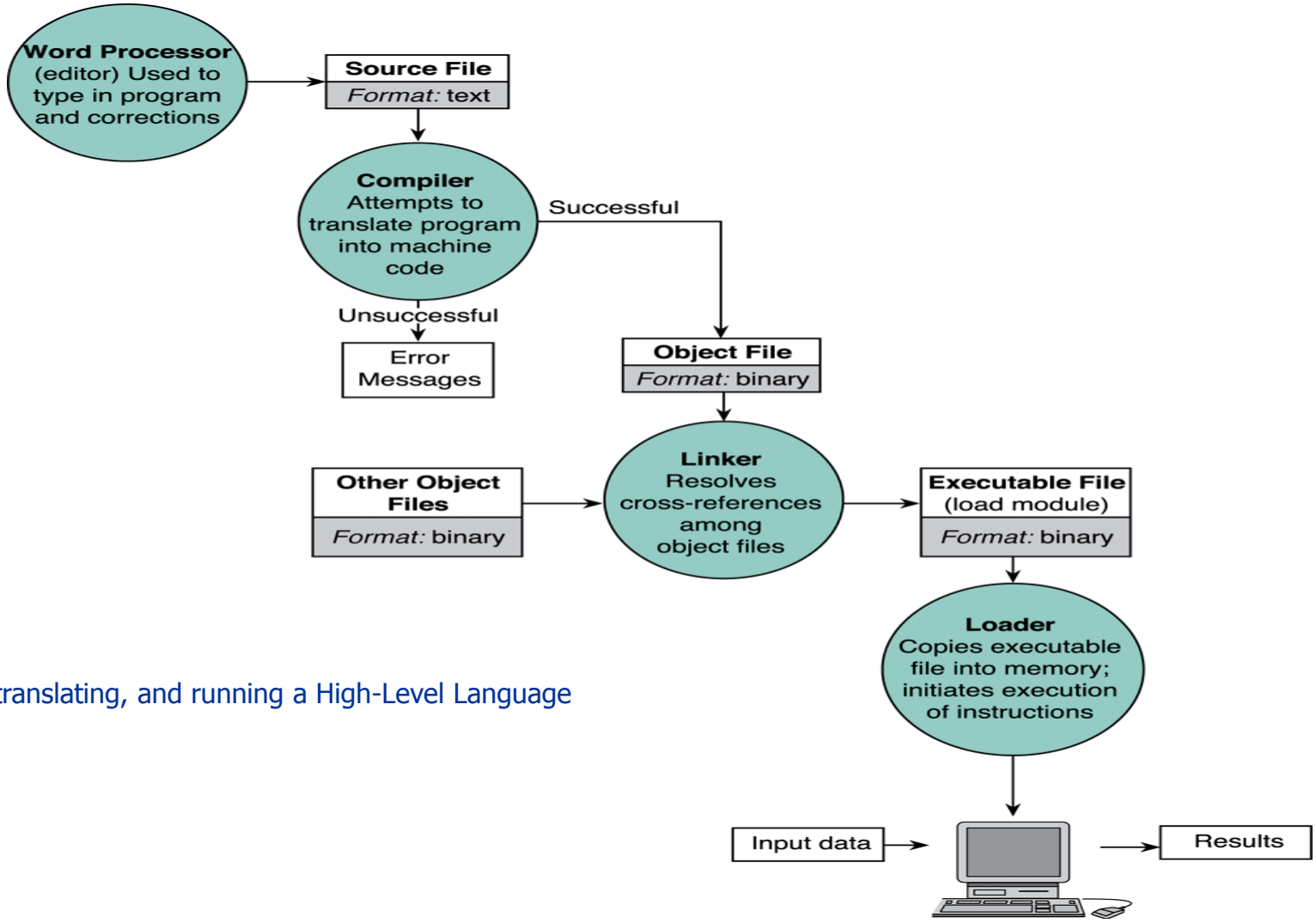
C Development Environment



C Development Environment



Entering, translating, and running a High-Level Language Program

Constant example – volume of a cone

```
#include <stdio.h>
int main(void)
{
    const double pi = 3.412;
    double height, radius, base, volume;

    printf("Enter the height and radius of the cone:");
    scanf("%lf %lf",&height, &radius);

    base = pi * radius * radius;
    volume = (1.0/3.0) * base * height;

    printf("\nThe volume of a cone is %f ", volume);
    return 0;
}
```

#define

- You may also associate constant using #define preprocessor directive

```
#include <stdio.h>
#define pi 3.412

int main(void)
{
    double height, radius, base, volume;

    printf("Enter the height and radius of the cone:");
    scanf("%lf %lf",&height,&radius);

    base = pi * radius * radius;
    volume = (1.0/3.0) * base * height;
    printf("\nThe volume of a cone is %f ", volume);
    return 0;
}
```

Escape Sequence

Escape Sequence	Effect
\a	Beep sound
\b	Backspace
\f	Formfeed (for printing)
\n	New line
\r	Carriage return
\t	Tab
\v	Vertical tab
\\	Backslash
\”	“ sign
\o	Octal decimal
\x	Hexadecimal
\O	NULL

Placeholder / Conversion Specifier

No	Conversion Specifier	Output Type	Output Example
1	%d	Signed decimal integer	76
2	%i	Signed decimal integer	76
3	%o	Unsigned octal integer	134
4	%u	Unsigned decimal integer	76
5	%x	Unsigned hexadecimal (small letter)	9c
6	%X	Unsigned hexadecimal (capital letter)	9C
7	%f	Integer including decimal point	76.0000
8	%e	Signed floating point (using e notation)	7.6000e+01
9	%E	Signed floating point (using E notation)	7.6000E+01
10	%g	The shorter between %f and %e	76
11	%G	The shorter between %f and %E	76
12	%c	Character	‘7’
13	%s	String	‘76'

Few notes on C program…

- C is ***case-sensitive***
  - Word, word, WorD, WORD, WOrD, worD, etc are all different variables / expressions

    Eg. sum = 23 + 7
    - What is the value of Sum after this addition ?
- Comments (remember 'Documentation'; Chapter 2)
  - are inserted into the code using /* to start and */ to end a comment
  - Some compiler support comments starting with '//'
  - Provides supplementary information but is ignored by the preprocessor and compiler
    - `/* This is a comment */`
    - `// This program was written by Hanly Koffman`

Few notes on C program cont…

- Reserved Words
  - Keywords that identify language entities such as statements, data types, language attributes, etc.
  - Have special meaning to the compiler, cannot be used as identifiers (variable, function name) in our program.
  - Should be typed in lowercase.
  - Example: const, double, int, main, void,printf, while, for, else (etc..)

Few notes on C program cont…

- Punctuators (separators)
  - Symbols used to separate different parts of the C program.
  - These punctuators include:

    [ ] ( ) { } , ; “: * #
  - Usage example:

```
int main void()
{
   int num = 10;
   printf(“%d”, num);
   return 0;
}
```

Common Programming Errors

- ***Debugging*** → Process removing errors from a program
- Three (3) kinds of errors :
  - Syntax Error
    - a violation of the C grammar rules, detected during program translation (compilation).
    - statement cannot be translated and program cannot be executed

Common Programming Errors

- Run-time errors
  - An attempt to perform an invalid operation, detected during program execution.
  - Occurs when the program directs the computer to perform an illegal operation, such as dividing a number by zero.
  - The computer will stop executing the program, and displays a diagnostic message indicates the line where the error was detected

Common Programming Errors

- Logic Error/Design Error
  - An error caused by following an incorrect algorithm
  - Very difficult to detect - it does not cause run-time error and does not display message errors.
  - The only sign of logic error – incorrect program output
  - Can be detected by testing the program thoroughly, comparing its output to calculated results
  - To prevent – carefully desk checking the algorithm and written program before you actually type it