Intermediate programming(C++)-

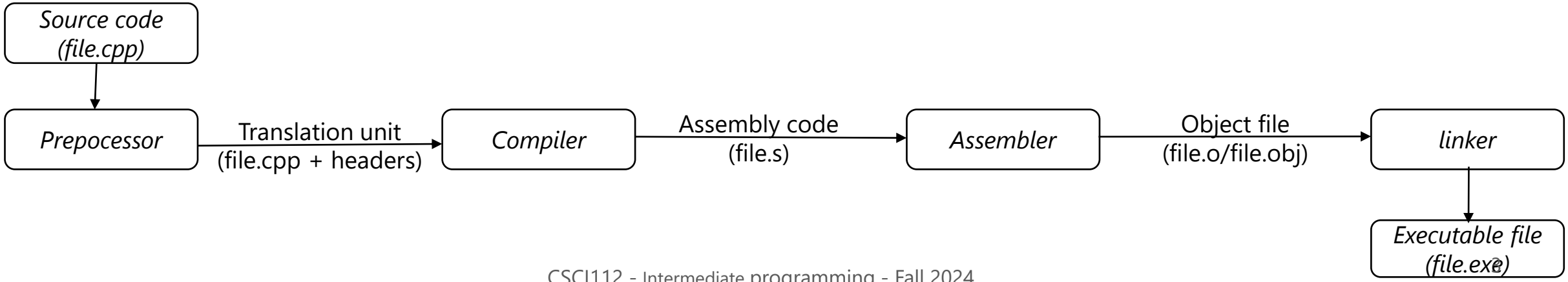
Lab 1

Content

- Introduction
- Your first C program
- Input/ Output
- Data types
- Examples
- Arithmetic operations
- Operator precedence
- Format specifiers
- Type casting
- Lab tasks

Introduction

- High level programming language
- ***Compiler***: Translates C/C++ code into binary/machine language, so the code can be executed.
- ***Applications***: Desktop applications/ Embedded systems/ Image processing
- ***C/C++ editors***: CBlocks/ Visual studio IDE/ Visual Studio code
- Only C++ supports Object-oriented programming (OOP).
- ***How C/C++ is executed?***



Your first C/C++ program

First C program:

```c
#include <stdio.h>

int main() {

    printf("Hello world!\n");

    return 0;

}
```

First C++ program:

```cpp
#include <iostream>
using namespace std;

int main() {

    Cout << "Hello world!\n";

    return 0;

}
```

Output:
Hello world!

Input/ Output

printf()/ cout <<

Console output

- To print specific statements or values stored in the memory
- Examples

```
// C programming language
Printf("hello reader\n");
Printf("hello reader\n");
Printf("hello reader \nwelcometo c programming language \n");

// C++ programming language
cout << "hello reader\n";
cout << "hello reader\n";
cout << "hello reader \nwelcometo c programming language \n";
```

Input/ Output

Scanf()/ cin >>

To take input from the user.

- Example

```c
// C programming
#include <stdio.h>

int main() {
    printf("EnterNo.");

    int num;
    scanf("%d", &num);
    printf("%d", num);

    return 0;
}
```

```cpp
// C++ programming
#include <iostream>
using namespace std;

int main() {
    printf("Enter No.");

    int num;
    cin >> num;
    cout << num;

    return 0;
}
```

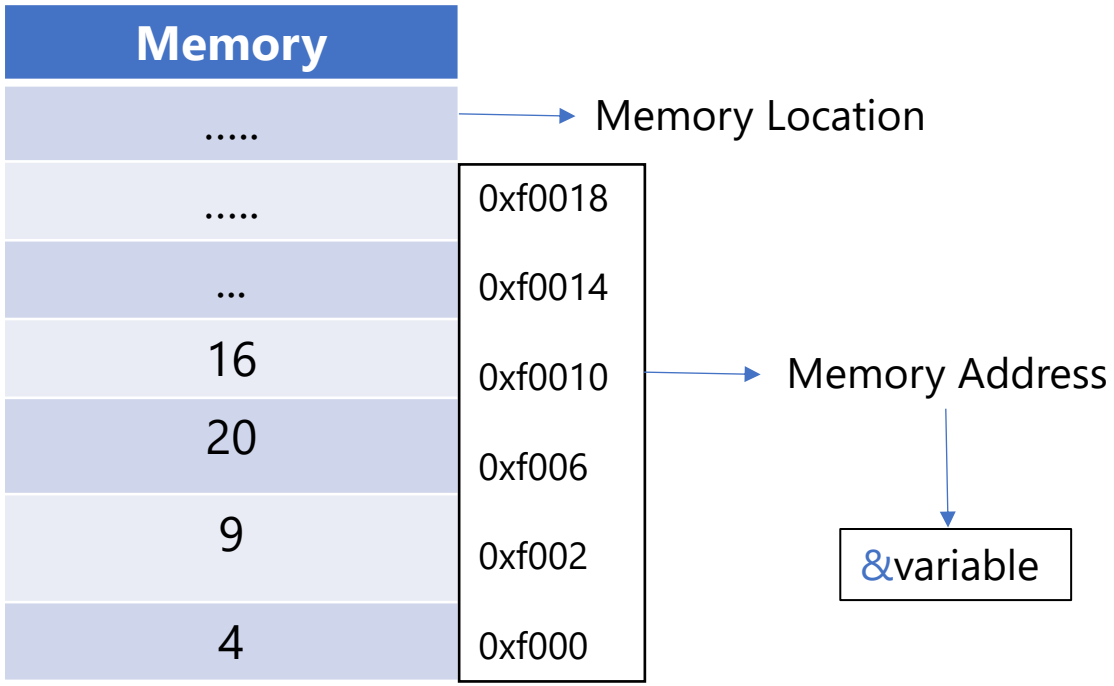
Input/ Output

& in scanf()

To get the address of the variable.

Example: scanf("%d", &num);

Means to get the integer number from the user and store it at the address of variable "num" where it is located in the memory.



Data types

Primary: int, float, char

- int (signed/unsigned)(2,4Bytes): used to store integers.
- char (signed/unsigned)(1Byte): used to store characters
- float, double(4,8Bytes): used to store a decimal number.

User Defined:

- typedef: used to rename a data type
- typedef int integer; can use integer to declare an int.
- enum, struct, union, class

Examples

```c
#include <stdio.h>

// C program prints a number of type int
int main() {
    int number = 4;
    printf ("Number is %d", number);

    return 0;
}
```

Output:
Number is 4

```c
#include <stdio.h>

// C program reads and prints the same thing
int main() {
    int number ;
    printf (" Enter a Number: ");
    scanf ("%d", &number);
    printf ("Number is %d\n", number);

    return 0;
}
```

Output
Enter a number: 4
Number is 4

Arithmetic operations

Prefix Increment: ++a

- *Example:*
  - int a=5;
  - b=++a; // value of b=6;a=6;

Postfix Increment: a++

- *Example:*
  - int a=5;
  - b=a++; //value of b=5; a=6;

Arithmetic operations

Modulus (remainder): %

- *Example:*
  - 12%5=2;

Assignment by addition:+=

- *Example:*
  - int a=4;
  - a+=1; //(means a=a+1) value of a becomes 5

Can use -, /, *, % also

Arithmetic operations

Comparison Operators: <, >, <=,>=, !=, ==, !,&&, ||

- *Example:*
  - int a=4, b=5;
  - a<b returns a true(nonzero number) value.

Bitwise Operators: <<, >>, ~, &|, ^

- *Example*
  - int a=8;
  - a=a>>1; // value of a becomes4

Examples

```c
// C program to add two numbers
#include <stdio.h>

int main(){
    int a= 4; //first number
    int b = 5; //second number
    int answer = 0; //result
    answer = a + b;
}
```

```cpp
// C++ program add two numbers
#include <iostream>
using namespace std;

int main(){
    int a= 4; //first number
    int b = 5; //second number
    int answer = 0; //result
    answer = a + b;
}
```

Operator precedence

- Meaning of a + b * c ?

  Is as a + (b * c) or (a + b) * c ?

- All operators have precedence over each other.
- *, / have more precedence over +, -
- If both *, / are used, associativity comes into picture.
- *Example*:

  5 + 4 * 3 = 5 +12 = 17

Highest on top
++ --(Postfix)
++ --(Prefix)
* / %
+ -
<< >>
< >
&
|
&&
||

Format specifiers

Example:

printf("%c", 'a'); scanf("%d", &a);

Format Specifier	Description
%c	For character type.
%d	For signed integer type.
%e or %E	For scientific notation of floats.
%f	For float type.
%g or %G	For float type with the current precision.
%i	signed integer
%ld or %li	Long
%lf	Double
%Lf	Long double
%lu	Unsigned int or unsigned long

%lli or %lld	Long long
%llu	Unsigned long long
%o	Octal representation
%p	Pointer
%s	String
%u	Unsigned int
%x or %X	Hexadecimal representation
%n	Prints nothing
%%	Prints % character

Type Casting

To change the variable's data type at specific statement in the program.

Example:

- `float x = 19.6;`
- `cout << (int) x;    // 19`

This means that x become 19.

Example2:

- `char z =  'a'`
- `cout << (int) z;    // 61`

Z is converted into 61 through ASCII table.

Lab tasks

- *Task 1:*

Write C statements to print the asterisk pattern as shown below.

*****

*****

*****

*****

*****

- *Task 2:*

Write a C program to take 2 integers from the user and calculate the product

- *Task 3:*

Write a C program to take 1int and1 float from the user, Calculate the sum once as int and once as float. Explain the output !