# **فصل 2**

## **کلمات کلیدی**

زبان سی پلاس پلاس کلمات کلیدی زیادی دارد برخی از آن‌ها به شرح زیر است.

1. If: ساختار شرطی برای اجرای بخشی از کد.

2. Else: بخشی از ساختار شرطی if-else.  
3. For: حلقه‌ی تکرار با شرط شروع، شرط ادامه، و گام تکرار.  
4.While : حلقه‌ی تکرار با شرط ادامه.  
5. Class: تعریف یک کلاس در برنامه.  
6.Function : تعریف یک تابع.

7. Return: بازگشت مقدار از یک تابع.

8. Namespace : فضای نام برنامه.

9. Cin و Cout: ورودی و خروجی استاندارد

10. Newو Delete: جهت مدیریت حافظه.

11. Auto: اتوماتیک تعیین نوع متغیر توسط کامپایلر.

12. Nullptr: مقدار پوینتر نال.

13. Static: استفاده در متغیرها و توابع برای تعریف قابل دسترسی در سطح فایل.

14. Const: مشخص کننده‌ی ثابت در تعریف متغیر یا تابع.

15. Template: امکان نوشتن کد ژنریک

16. Try و Catch و Throw: برنامه‌نویسی استثناء برای مدیریت خطا.

17. Virtual: استفاده در تعریف توابع و کلاس‌ها برای پلی‌مورفیسم.

18. Friend: اجازه به یک تابع یا کلاس برای دسترسی به اعضای private یک کلاس.

19. Break و Continue: کنترل حلقه‌ها و سوئیچ‌ها.

20. This: پوینتر به شیء فعلی درون یک عضو تابع کلاس.

21. Public: دسترسی به اعضای کلاس خارج از آن  
22. Private: دسترسی به اعضای کلاس فقط در همان کلاس

23.Protected : مانند پرایوت ولی زیرکلاس‌ها نیز به ان دسترسی دارند

24. Inline: برای تعریف تابع‌هایی که به صورت مستقیم در خطوط کد فراخوانی می‌شوند و از اجرای کد متنی جلوگیری کنند.

25. Final: جلوگیری از ارث‌بری کلاس توسط دیگر کلاس‌ها.

26. Override: نشان‌دهنده‌ی ارث‌بری تابع واقعی در کلاس مشتق‌شده.

27. Dynamic\_cast: تبدیل پوینترها یا مراجع در زمان اجرا با استفاده از پلی‌مورفیسم.

28. Enum: تعریف یک مجموعه ثابت از مقادیر.

29. Volatile: اطلاع‌رسانی به کامپایلر که مقدار یک متغیر ممکن است تغییر کند.

30. Smart pointers: اشاره‌گرهای هوشمند برای مدیریت حافظه بهتر.

## **گرامر و تجزیه درخت**

گرامر برای حلقه while

while-loop → while ( condition ) statement  
  
condition  → expression  
  
statement  → simple-statement | compound-statement  
  
simple-statement → expression-statement  
                | variable-declaration  
  
compound-statement → { statement-list }  
  
statement-list → statement | statement-list statement  
  
expression-statement → expression ;  
  
variable-declaration → type-specifier identifier ;  
  
expression → additive-expression  
  
additive-expression → multiplicative-expression | additive-expression + multiplicative-expression | additive-expression - multiplicative-expression  
  
multiplicative-expression → primary-expression | multiplicative-expression \* primary-expression | multiplicative-expression / primary-expression  
  
primary-expression → identifier | constant | ( expression )

گرامر برای تعریف رفرنس

<program> ➝ <declaration\_list>  
  
<declaration\_list> ➝ <declaration> | <declaration><declaration\_list>  
  
<declaration> ➝ <variable\_declaration> | <reference\_assignment\_declaration>| <assign\_declaration>  
  
<variable\_declaration> ➝ <type> <letter> ";"  
  
<reference\_assignment\_declaration> ➝ <type> "&" <letter> "=" <letter> ";"

<assign\_declaration>➝ <letter> "=" <letter> ";"|<letter> "=" <number> ";"

<type> ➝ "int"|"double" | "char"|"bool"|"void"|….  
  
<number> ➝ <digit> | <number> <digit>  
  
<letter> ➝ "a" | "b" | ... | "z" | "A" | "B" | ... | "Z"  
  
<digit> ➝ "0" | "1" | "2" | ... | "9"

یک قطعه کد ساده داریم:

int main() {  
    int A;  
    int& B = A;  
    B = 42;

}

درخت تجزیه مطابق زیر خواهد بود:

**program**

**Declaration\_list**

**Declaration**

**Declaration\_list**

**Declaration**

**Declaration\_list**

**Variable\_declaration**

**Declaration**

**reference\_assignment\_declaration**

**;"**"

**letter**

**type**

**assign\_declaration**

**type**

**"&**"

**letter**

**="**"

**letter**

**;"**"

**"A"**

**Int"**"

**"B"**

**Int"**"

**number**

**="**"

**letter**

**"A"**

**"B"**

**;"**"

**"2"**

**"4"**

**digit**

**digit**

**number**