



Jacoms and 8	Anline Function.				
· · · · · · · · · · · · · · · · · · ·					
0 eliminate to	ime overneool.				
lacros:					
Je nud # olifi	ms compiler divu	ctive and	lit û wud for text alleafitu	hon	
imtar: # deli	M. Sougre (X	·) (2.*	~ √√)		
•	•				
briuth 2.00	, equare(5));	expano	470 5 * 5,		
Nd vanto e		di advantane			
· ·		_			
MBC I I GOOD OF (.					
	wing inline	. Jeyd	word and in mudito opti	mize function calls	
-Court Park	ا محمد الحدث وسالًا	° 0 +	1 C madurus 2 2027.		
				. 1.7	
printf(")	/· ot", aquare (S) ا ا / ا	y be obtimized by the con	npiler	
D.domise a			ditadvantese		
aide e	effects		inline auggestis		
oth aim to suc	duce l unction ca	ell Over	thead but diller in implem	nentation.	
Feature	Macro		Inline Function		
Definition	Preprocessor directive (#def	ine)	Function definition (inline keyword)		
Type Safety	No type checking		Type-safe, function-like behavior		
Compilation	Handled by preprocessor		Handled by the compiler		
Debugging	Hard to debug (text substitu	tion)	Easier to debug (like normal functions)		
Side Effects	Possible due to multiple eva	luations	No side effects, behaves like a function		
Code Bloat	Can cause code duplication		Compiler optimizes inline code		
1	De use them as to eliminate to eliminate to eliminate to eliminate to enud # oldinate to eliminate	Coliminate time overhead. Cacros: Je nual # olyine compiler dire yntax: # define square (x printf(" y-d", square(5)); Advantge tid fast and quick text whatitution. Inline functions His defined using inline by the compiler Syntax: inline intaquare (printf(" y-d", square (S) Advantge Type adjety, debuggable, r aide effects Oth aim to reduce function co Feature Macro Definition Preprocessor directive (#def Type Safety No type checking Compilation Handled by preprocessor Debugging Hard to debug (text substitut Side Effects Possible due to multiple eva	De use them when sequence of ately is O eliminate time overhead. Octos: De need # olime compiler directive and yntax: # define square(x) (xx printf(" y. ol", equare(5)); expans Advantge tid fact and quick text No whatilution. Inline functions Of is defined using inline keye by the compiler Syntax: inline intaquare (intax printf(" y. ol", aquare (5); // Ma Advantge Type adjety, debuggable, no aide effects Oth aim to reduce function call over Feature Macro Definition Preprocessor directive (#define) Type Safety No type checking Compilation Handled by preprocessor Debugging Hard to debug (text substitution) Side Effects Possible due to multiple evaluations	De sus them when sequence of deeps is small O climinate time overhead. Octions time compiler directors and it is used for text alleast time. Octions time square (x) (x * x) Printf(" y. ol". aquare(s)); expands to 5 * 5, No type checking. Octions time time time dependence where times times times to the same times times times. Octions times introduce (int x) { xuturn x * x x }; Printf(" y. ol". aquare (s); // Naybe obtainized by the Constitution of times to reduce function call overhead but differ in implementation. Oth aim to reduce function call overhead but differ in implementation. Octions to reduce function call overhead but differ in implementation. Octions to reprocessor directive (meeting) Octions time to reduce function times keyword) Type Safety No type checking Compilation Handled by preprocessor Debugging Hard to debug (text substitution) Side Effects. Possible due to multiple evaluations No side effects, behaves like a function in the second of times to debug (like normal functions) No side effects, behaves like a function.	

Macros

**Definition **: Macros are preprocessor directives in C/C++ that define a set of code that can be reused. They are typically defined using the `#define` directive.

- **Advantages**:
- 1. **Performance**: Since macros are expanded inline during preprocessing, there is no function call overhead.

 2. **Code Reusability**: Allows you to write a piece of code once and use it multiple
- times.

 3. **Conditional Compilation**: Macros can be used to include or exclude parts of code based on certain conditions.
- **Disadvantages**:
- 1. **Debugging Difficulty**: Errors in macros can be hard to trace, as the code expanded from macros may not show the original macro definition.
 2. **No Type Checking**: Macros do not check types, which can lead to unexpected
- behavior.

 3. **Side Effects**: If a macro is defined with arguments that have side effects (like
- 3. **Side Effects**: If a macro is defined with arguments that have side effects (like function calls), it can lead to unexpected results when the macro is expanded.
- ### Inline Functions
- **Definition**: Inline functions are functions that are defined with the `inline` keyword in C/C++. They suggest to the compiler to insert the function's code at each point the function is called, instead of performing a traditional call.
- **Advantages**:

 1 **Type Safety**: Inline functions provide type
 - 1. **Type Safety**: Inline functions provide type checking, helping to avoid errors that macros might introduce.
 - 2. **Debugging Ease**: Errors in inline functions can be traced back to their definitions, making debugging easier.
 - 3. **Scoped Variables**: Inline functions can have local variables, avoiding potential conflicts seen with macros.
 - **Disadvantages**:
 - 1. **Code Bloat**: If an inline function is used many times, it can increase the size of the binary since the code is duplicated.
 - 2. **Compiler Discretion**: The compiler may ignore the inline request and treat it like a regular function, especially if the function is complex or large.
 - 3. **Overhead for Small Functions**: For very small functions, the overhead of making the call can sometimes outweigh the benefits of inlining.
 - ### Summary

Feature	Macros	Inline Functions	
Definition	Preprocessor directives for code reuse.	Functions defined with the inline keyword.	
Advantages	Performance, code reusability, conditional compilation.	Type safety, debugging ease, scoped variables.	
Disadvantages Debugging difficulty, no type checking, potential side effects.		Code bloat, compiler discretion, overhead for small functions.	