1. What is function prototyping in C++?
   1. **Declaring a function before it is defined**
   2. Defining a function before it is declared
   3. Declaring a function after it is defined
   4. None of the above
2. Which of the following is the correct syntax for function prototyping?
   1. **return-type function-name(parameters);**
   2. return-type parameters function-name;
   3. function-name(parameters) return-type;
   4. None of the above
3. Function overloading in C++ allows:
   1. Defining multiple functions with the same name but different return types
   2. Defining multiple functions with the same name and the same return type
   3. **Defining multiple functions with the same name but different parameters**
   4. None of the above
4. What is required for function overloading to work in C++?
   1. Functions must have the same name and different return types
   2. Functions must have the same name and the same return type
   3. **Functions must have the same name but different parameters**
   4. Functions must have the same name but different access specifiers
5. Which of the following is an example of function overloading?
   1. **int add(int a, int b);**

**int add(float a, float b);**

* 1. int add(int a, int b);

int add(int a, int c);

* 1. int add(int a, int b);

float add(int a, int b);

* 1. None of the above

1. Inline functions in C++ are expanded by the compiler:
   1. **At the function call site**
   2. At the function definition site
   3. At the function prototype site
   4. None of the above
2. Which of the following is an advantage of using inline functions in C++?
   1. Improved runtime performance
   2. Reduced code size
   3. Faster execution
   4. **All of the above**
3. Which keyword is used to define an inline function in C++?
   1. **inline**
   2. def
   3. func
   4. inlinefunc
4. The decision of whether to inline a function is made by:
   1. The programmer
   2. **The compiler**
   3. The linker
   4. The preprocessor
5. Which of the following is a drawback of using inline functions in C++?
6. **Increased code size**
7. Reduced runtime performance
8. Difficulty in debugging
9. All of the above
10. Which of the following is an example of an inline function?
    1. int square(int num) { return num \* num; }

**b) int add(int a, int b) { return a + b; }**

c) int factorial(int num);

inline int factorial(int num) { /\* function definition \*/ }

d) None of the above

1. What is the purpose of function prototyping?
   1. To specify the return type of a function
   2. To specify the parameters of a function
   3. **To declare a function before it is defined**
   4. All of the above
2. Which of the following is true about function prototyping in C++?
   1. **It is optional**
   2. It is mandatory
   3. It is required only for recursive functions
   4. None of the above
3. Which section of a C++ program typically contains function prototypes?
   1. Global section
   2. Main section
   3. Local section
   4. **Header section**
4. What is the purpose of function overloading in C++?
   1. To reduce code duplication
   2. To improve code readability
   3. To provide multiple ways to call a function
   4. **All of the above**
5. Can function overloading be based solely on the return type of the function?
   1. Yes
   2. **No**
6. Which of the following is true about the parameters of overloaded functions in C++?
   1. They must have the same number of parameters
   2. They must have the same types of parameters
   3. They can have different order of parameters
   4. **All of the above**
7. Can a function be overloaded based on the constness of its parameters in C++?
   1. **Yes**
   2. No
8. Which of the following is true about the return type of overloaded functions in C++?
   1. **The return type can be different for overloaded functions**
   2. The return type must be the same for overloaded functions
   3. The return type is not considered for function overloading
   4. None of the above
9. Which keyword is used to specify that a function is to be inlined in C++?
   1. **inline**
   2. def
   3. func
   4. static
10. What happens if the inline keyword is used with a recursive function in C++?
    1. **The function is not inlined**
    2. The function is inlined with recursion
    3. The function is inlined without recursion
    4. None of the above
11. Which section of a C++ program contains the function definitions?
    1. **Global section**
    2. Main section
    3. Local section
    4. All of the above
12. Which keyword is used to define a function in C++?
    1. function
    2. def
    3. func
    4. **None of the above**
13. What is the purpose of function prototypes in C++?
    1. **To provide a forward declaration of a function**
    2. To define the implementation of a function
    3. To declare the return type of a function
    4. None of the above
14. What is required for function overloading to work in C++?
    1. Functions must have the same name and different return types
    2. Functions must have the same name and the same return type
    3. **Functions must have the same name but different parameters**
    4. Functions must have the same name but different access specifiers
15. Which keyword is used to define a function in C++ when overloading functions?
    1. function
    2. def
    3. func
    4. **None of the above**
16. What happens if a function is used in C++ without a function prototype or definition?
    1. **It will cause a compilation error**
    2. It will automatically be treated as an inline function
    3. It will be interpreted as a function with no return type
    4. None of the above
17. What is the syntax for function prototyping in C++?
    1. **return-type function-name(parameters);**
    2. return-type parameters function-name;
    3. function-name(parameters) return-type;
    4. None of the above