1. Which of the following is a correct identifier in C++?  
a) 7var\_name  
b) 7VARNAME  
c) VAR\_1234  
d) $var\_name

Answer: c  
Explanation: The rules for writing an identifier is as follows:  
i) may contain lowercase/uppercase letters, digits or underscore(\_) only  
ii) should start with a non-digit character  
iii) should not contain any special characters like @, $, etc.

2. Which of the following is called address operator?  
a) \*  
b) &  
c) \_  
d) %

Answer: b  
Explanation: & operator is called address operator and is used to access the address of a variable.

3. Which of the following is used for comments in C++?  
a) // comment  
b) /\* comment \*/  
c) both // comment or /\* comment \*/  
d) // comment \*/

Answer: c  
Explanation: Both the ways are used for commenting in C++ programming. // is used for single line comments and /\* … \*/ is used for multiple line comments.

4. Which function is used to read a single character from the console in C++?  
a) cin.get(ch)  
b) getline(ch)  
c) read(ch)  
d) scanf(ch)

Answer: a  
Explanation: C++ provides cin.get() function to read a single character from console whereas others are used to read either a single or multiple characters.

5. Which function is used to write a single character to console in C++?  
a) cout.put(ch)  
b) cout.putline(ch)  
c) write(ch)  
d) printf(ch)

Answer: a  
Explanation: C++ provides cout.put() function to write a single character to console whereas others are used to write either a single or multiple characters.

6. What are the escape sequences?  
a) Set of characters that convey special meaning in a program  
b) Set of characters that whose use are avoided in C++ programs  
c) Set of characters that are used in the name of the main function of the program  
d) Set of characters that are avoided in cout statements

Answer: a  
Explanation: Escape sequence is a set of characters that convey a special meaning to the program. They are used to convey a meaning which cannot be conveyed directly.

7. Which of the following escape sequence represents carriage return?  
a) \r  
b) \n  
c) \n\r  
d) \c

Answer: a  
Explanation: \r is used to represent carriage return which means move the cursor to the beginning of the next line.

8. Which of the following escape sequence represents tab?  
a) \t  
b) \t\r  
c) \b  
d) \a

Answer: a  
Explanation: \t is used to represent tab which means a set of blank spaces in a line.

9. Which of the following is called insertion/put to operator?  
a) <<  
b) >>  
c) >  
d) <

Answer: a  
Explanation: << operator is called insertion or put to operator i.e. insert/put things to console/files.

10. Which of the following is called extraction/get from operator?  
a) <<  
b) >>  
c) >  
d) <

Answer: b  
Explanation: >> operator is called extraction or get from operator i.e. extract/get things from console/files.

11. Which of the following is not a fundamental type is not present in C but present in C++?  
a) int  
b) float  
c) bool  
d) void

Answer: c  
Explanation: Boolean type is not present as a fundamental type in C. int type is used as boolean in C whereas in C++ bool is defined as a fundamental type for handling boolean outputs.

12. What is the size of a boolean variable in C++?  
a) 1 bit  
b) 1 byte  
c) 4 bytes  
d) 2 bytes

Answer: a  
Explanation: Boolean uses only 1 bit as it stores only truth values which can be true(1) or false(0).

13. Which of the following is C++ equivalent for scanf()?  
a) cin  
b) cout  
c) print  
d) input

Answer: a  
Explanation: C++ uses cin to read input form uses. However C++ also uses scanf().

14. Which of the following is C++ equivalent for printf()?  
a) cin  
b) cout  
c) print  
d) input

Answer: b  
Explanation: C++ uses cout to print output to console. However C++ also uses printf().

15. Which of the following is the correct difference between cin and scanf()?  
a) both are the same  
b) cin is a stream object whereas scanf() is a function  
c) scanf() is a stream object whereas cin is a function  
d) cin is used for printing whereas scanf() is used for reading input

Answer: b  
Explanation: cin is a stream object available in C++ whereas scanf() is a function available in both C and C++. both are used for reading input from users.

16. In which part of the for loop termination condition is checked?  
for(I;II;III)  
{IV}  
a) I  
b) II  
c) III  
d) IV

Answer: b  
Explanation: In II part the termination condition of the for loop is checked.

17. What is dynamic binding?  
a) The process of linking the actual code with a procedural call during run-time  
b) The process of linking the actual code with a procedural call during compile-time  
c) The process of linking the actual code with a procedural call at any-time  
d) All of the mentioned

Answer: a  
Explanation: Binding of calls and variables with actual code at run-time is called dynamic binding. For example in the concept of polymorphism types are decided are defined during the execution of code which leads to the different function calls depending upon the types used this is called dynamic binding. As the function call is decided during the run-time therefore dynamic binding happens at run-time.

18.  What is static binding?  
a) The process of linking the actual code with a procedural call during run-time  
b) The process of linking the actual code with a procedural call during compile-time  
c) The process of linking the actual code with a procedural call at any-time  
d) All of the mentioned

Answer: b  
Explanation: Binding of calls and variables with actual code at compile-time is called static binding. For example normally whenever we declare a variable we define its type hence compiler knows what type should be binded to that variable i.e. compiler can decide about that variable this is called static binding.

19. Which of the following is the scope resolution operator?  
a) .  
b) \*  
c) ::  
d) ~

Answer: c  
Explanation: :: operator is called scope resolution operator used for accessing a global variable from a function which is having the same name as the variable declared in the function.

20.  What will be the output of the following C++ code?

#include<iostream>

using namespace std;

int x = 1;

int main()

{

int x = 2;

{

int x = 3;

cout << ::x << endl;

}

return 0;

}

a) 1  
b) 2  
c) 3  
d) 123

Answer: a  
Explanation: While printing x we are using :: operator hence the reference is given to global variable hence the global variable x = 1 is printed.

21. What will be the output of the following C++ code?

#include<iostream>

using namespace std;

int x[100];

int main()

{

cout << x[99] << endl;

}

a) Garbage value  
b) 0  
c) 99  
d) Error

Answer: b  
Explanation: In C++ all the uninitialized variables are set to 0 therefore the value of all elements of the array is set to 0.

22. Which of the following is accessed by a member function of a class?  
a) The object of that class  
b) All members of a class  
c) The public part of a class  
d) The private part of a class

Answer: b  
Explanation: A member function of a class can access all the members of its class whether they are private, protected or public.

23. Which of the following is correct?  
a) struct cannot have member function in C but it can in C++  
b) struct cannot have member function in C++ but it can in C  
c) struct cannot have member function in both C and C++  
d) struct can have member function in both C and C++

Answer: a  
Explanation: struct can have member function in C++ whereas member functions are not allowed in case of C.

24. What happens if we run the following code in both C and C++?

#include<stdio.h>

struct STRUCT

{

int a;

int func()

{

printf("HELLO THIS IS STRUCTURE**\n**");

}

};

int main()

{

struct STRUCT s;

s.func();

return 0;

}

a) The program runs fine and both prints output “HELLO THIS IS STRUCTURE”  
b) The program gives an error in case of C but runs perfectly in case of C++  
c) The program gives an error in case of C++ but runs perfectly in case of C  
d) The program gives an error in case of both C and C++

Answer: b  
Explanation: As C does not allows the structure to have member functions, therefore, it gives an error in case of C but as C++ does allow structures to have member functions, therefore, the C++ does not give an error.

25. What happens if the following program is compiled in both C and C++?

#include<stdio.h>

struct STRUCT

{

int static a;

};

int main()

{

struct STRUCT s;

return 0;

}

a) The program runs fine and both prints output “HELLO THIS IS STRUCTURE”  
b) The program gives an error in case of C but runs perfectly in case of C++  
c) The program gives an error in case of C++ but runs perfectly in case of C  
d) The program gives an error in case of both C and C++

Answer: b  
Explanation: C does not allow the programmer to declare any static members inside a class whether in C++ it is allowed to declare static variables.

26. Which of the following statement is correct?  
a) Structure in C allows Constructor definition  
b) Structure in C++ allows Constructor definition  
c) Both allow Constructor definition  
d) C allows constructor definition while C++ does not

Answer: b  
Explanation: As C does not allow the programmer to define a function inside a structure and constructor itself is a function, therefore, the constructor definition is not allowed in C whereas such definitions are allowed in C++.

27. What happens if the following code is compiled on both C and C++?

#include<stdio.h>

struct STRUCT

{

private:

int a;

};

int main()

{

printf("%d**\n**", (int)sizeof(struct STRUCT));

return 0;

}

a) The program runs fine and both prints output “HELLO THIS IS STRUCTURE”  
b) The program gives an error in case of C but runs perfectly in case of C++  
c) The program gives an error in case of C++ but runs perfectly in case of C  
d) The program gives an error in case of both C and C++

Answer: b  
Explanation: Access specifiers like private, protected and the public are used because the OOPs concept and as C is not an Object Oriented language, therefore, access specifiers are not defined in C and hence C gives error whereas C++ does not.

28. Which of the following operator is used with this pointer to access members of a class?  
a) .  
b) !  
c) ->  
d) ~

Answer: c  
Explanation: this pointer is a type of pointer and as we know pointer object uses the arrow(->) operator to access the members of the class, therefore, this pointer uses -> operator.

29. Why this pointer is used?  
a) To access the members of a class which have the same name as local variables in that scope  
b) To access all the data stored under that class  
c) To access objects of other class  
d) To access objects of other variables

Answer: a  
Explanation: this pointer is used to access the members of a class which have the same name as local variables in that part of the code.

30. What is std in C++?  
a) std is a standard class in C++  
b) std is a standard namespace in C++  
c) std is a standard header file in C++  
d) std is a standard file reading header in C++

Answer: b  
Explanation: std is a standard namespace present in C++ which contains different stream classes and objects like cin, cout, etc. and other standard functions.

31. What will be the output of the following C++ code?

#include <iostream>

int main(int argc, char const \*argv[])

{

cout<<"Hello World";

return 0;

}

a) Hello World  
b) Compile-time error  
c) Run-time error  
d) Segmentation fault

Answer: b  
Explanation: cout is defined under the namespace std and without including std namespace we cannot cout, therefore, the program gives an error.

32. Which of the following syntax can be used to use a member of a namespace without including that namespace?  
a) namespace::member  
b) namespace->member  
c) namespace.member  
d) namespace~member

Answer: a  
Explanation: To use a member of a namespace without including the namespace is done by this syntax namespace::member.

33. Which of the following C++ code will give error on compilation?

================code 1=================

#include <iostream>

using namespace std;

int main(int argc, char const \*argv[])

{

cout<<"Hello World";

return 0;

}

========================================

================code 2=================

#include <iostream>

int main(int argc, char const \*argv[])

{

std::cout<<"Hello World";

return 0;

}

========================================

a) Both code 1 and code 2  
b) Code 1 only  
c) Code 2 only  
d) Neither code 1 nor code 2

Answer: d  
Explanation: Neither code 1 nor code2 will give error as both are syntactically correct as in first code we have included namespace std and in second one we have used scope resolution operator to resolve the conflict.