Mcq QUESTIONS

1. Who invented C++?  
   a) Dennis Ritchie  
   b) Ken Thompson  
   c) Brian Kernighan  
   d) Bjarne Stroustrup

Answer: d  
Explanation: Bjarne Stroustrup is the original creator of C++ in 1979 at AT&T Bell Labs.

1. What is C++?  
   a) C++ is an object-oriented programming language  
   b) C++ is a procedural programming language  
   c) C++ supports both procedural and object-oriented programming language  
   d) C++ is a functional programming language

Answer: c  
Explanation: C++ supports both procedural(step by step instruction) and object oriented programming (using the concept of classes and objects).

1. Which of the following is the correct syntax of including a user defined header files in C++?  
   a) #include [userdefined]  
   b) #include “userdefined”  
   c) #include <userdefined.h>  
   d) #include <userdefined>

Answer: b  
Explanation: C++ uses double quotes to include a user-defined header file. The correct syntax of including user-defined is #include “userdefinedname”

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

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

5. Which of the following user-defined header file extension used in c++?  
a) hg  
b) cpp  
c) h  
d) hf  
Answer: c  
Explanation: .h extensions are used for user defined header files. To include a user defined header file one should use #include”name.h” i.e. enclosed within double quotes

advertisement

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

Answer: a  
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.

7. Which of the following is not a type of Constructor in C++?  
a) Default constructor  
b) Parameterized constructor  
c) Copy constructor  
d) Friend constructor

Answer: d  
Explanation: Friend function is not a constructor whereas others are a type of constructor used for object initialization.

8. Which of the following approach is used by C++?  
a) Left-right  
b) Right-left  
c) Bottom-up  
d) Top-down

Answer: c  
Explanation: C++ is an object-oriented language and OOL uses a bottom-up approach to solve/view a problem.

9. What is virtual inheritance in C++?  
a) C++ technique to enhance multiple inheritance  
b) C++ technique to ensure that a private member of the base class can be accessed somehow  
c) C++ technique to avoid multiple inheritances of classes  
d) C++ technique to avoid multiple copies of the base class into children/derived class

Answer: d  
Explanation: Virtual inheritance is a C++ technique with which it ensures that a derived class contains only one copy of the base class’s variables. Refer Wikipedia for more info.

10. What happens if the following C++ statement is compiled and executed?

int \*ptr = NULL;

delete ptr;

a) The program is not semantically correct  
b) The program is compiled and executed successfully  
c) The program gives a compile-time error  
d) The program compiled successfully but throws an error during run-time

Answer: b  
Explanation: The above statement is syntactically and semantically correct as C++ allows the programmer to delete a NULL pointer, therefore, the program is compiled and executed successfully.

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

#include <iostream>

#include <string>

using namespace std;

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

{

char s1[6] = "Hello";

char s2[6] = "World";

char s3[12] = s1 + " " + s2;

cout<<s3;

return 0;

}

a) Hello  
b) World  
c) Error  
d) Hello World  
Answer: c  
Explanation: There is no operation defined for the addition of character array in C++ hence the compiler throws an error as it does not understood what to do about this expression.

12. What is the difference between delete and delete[] in C++?  
a) delete is syntactically correct but delete[] is wrong and hence will give an error if used in any case  
b) delete is used to delete normal objects whereas delete[] is used to pointer objects  
c) delete is a keyword whereas delete[] is an identifier  
d) delete is used to delete single object whereas delete[] is used to multiple(array/pointer of) objects

Answer: d  
Explanation: delete is used to delete a single object initiated using new keyword whereas delete[] is used to delete a group of objects initiated with the new operator.

13. What happens if the following program is executed in C and C++?

#include <stdio.h>

int main(void)

{

int new = 5;

printf("%d", new);

}

a) Error in C and successful execution in C++  
b) Error in both C and C++  
c) Error in C++ and successful execution in C  
d) A successful run in both C and C++

Answer: c  
Explanation: new is a keyword in C++, therefore, we cannot declare a variable with name new but as there is no such keyword new in C, therefore, the program is compiled and executed successfully in C.

14. What happens if the following program is executed in C and C++?

#include <stdio.h>

void func(void)

{

printf("Hello");

}

void main()

{

func();

func(2);

}

a) Outputs Hello twice in both C and C++  
b) Error in C and successful execution in C++  
c) Error in C++ and successful execution in C  
d) Error in both C and C++

Answer: d  
Explanation: As the func(void) needs no argument during its call, hence when we are calling func(2) with 2 as passed as a parameter then this statement gives the error in both C++ and C compiler.

15. Which of the following is correct about this pointer in C++?  
a) this pointer is passed as a hidden argument in all static variables of a class  
b) this pointer is passed as a hidden argument in all the functions of a class  
c) this pointer is passed as a hidden argument in all non-static functions of a class  
d) this pointer is passed as a hidden argument in all static functions of a class  
Answer: c  
Explanation: As static functions are a type of global function for a class so all the object shares the common instance of that static function whereas all the objects have there own instance for non-static functions and hence they are passed as a hidden argument in all the non-static members but not in static members.

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

1. #include <iostream>
2. #include <string>
3. #include <algorithm>
4. using namespace std;
5. int main()
6. {
7. string s = "spaces in text";
8. s.erase(remove(s.begin(), s.end(), ' ' ), s.end() ) ;
9. cout << s << endl;
10. }

a) spacesintext  
b) spaces in text  
c) spaces  
d) spaces in

Answer: a  
Explanation: In this program, We formed a algorithm to remove spaces in the string.  
Output:

$ g++ dan.cpp

$ a.out

spacesintext

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

Answer: b  
Explanation: Neither code 1 nor code 2 will give an 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.

18. Which of the following type is provided by C++ but not C?  
a) double  
b) float  
c) int  
d) bool  
Answer: d  
Explanation: C++ provides the boolean type to handle true and false values whereas no such type is provided in C.

19. What is the value of p in the following C++ code snippet?

1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5. int p;
6. bool a = true;
7. bool b = false;
8. int x = 10;
9. int y = 5;
10. p = ((x | y) + (a + b));
11. cout << p;
12. return 0;
13. }

a) 12  
b) 0  
c) 2  
d) 16  
Answer: d  
Explanation: | means bitwise OR operation so x | y (0101 | 1010) will be evaluated to 1111 which is integer 15 and as a is true and b is false so a+b(1 + 0) = 1. So final value of expression in line #10 will be 15 + 1 = 16.

20. By default, all the files in C++ are opened in \_\_\_\_\_\_\_\_\_ mode.  
a) Binary  
b) VTC  
c) Text  
d) ISCII  
Answer: c  
Explanation: By default, all the files in C++ are opened in text mode. They read the file as normal text.

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

1. int main()
2. {
3. register int i = 1;
4. int \*ptr = &i;
5. cout << \*ptr;
6. return 0;
7. }

a) Runtime error may be possible  
b) Compiler error may be possible  
c) 1  
d) 0  
Answer: b  
Explanation: Using & on a register variable may be invalid, since the compiler may store the variable in a register, and finding the address of it is illegal.

22. Which of the following correctly declares an array in C++?  
a) array{10};  
b) array array[10];  
c) int array;  
d) int array[10];  
Answer: d  
Explanation: Because array variable and values need to be declared after the datatype only.

23. What is the size of wchar\_t in C++?  
a) Based on the number of bits in the system  
b) 2 or 4  
c) 4  
d) 2  
Answer: a  
Explanation: Compiler wants to make CPU as more efficient in accessing the next value.

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

#include<iostream>

using namespace std;

int main ()

{

int cin;

cin >> cin;

cout << "cin: " << cin;

return 0;

}

a) Segmentation fault  
b) Nothing is printed  
c) Error  
d) cin: garbage value  
Answer: d  
Explanation: cin is a variable hence overrides the cin object. cin >> cin has no meaning so no error.

25. What is the use of the indentation in c++?  
a) r distinguishes between comments and inner data  
b) distinguishes between comments and outer data  
c) distinguishes between comments and code  
d) r distinguishes between comments and outer data  
Answer: c  
Explanation: To distinguish between different parts of the program like comments, codes, etc.

26. Which is more effective while calling the C++ functions?  
a) call by object  
b) call by pointer  
c) call by value  
d) call by reference  
Answer: d  
Explanation: In the call by reference, it will just passes the reference of the memory addresses of passed values rather than copying the value to new memories which reduces the overall time and memory use.

27. What will be the output of the following C++ program?

#include <iostream>

#include <string>

#include <cstring>

using namespace std;

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

{

const char \*a = "Hello**\0**World";

cout<<a;

return 0;

}

a) Hello  
b) World  
c) Error  
d) Hello World  
Answer: a  
Explanation: char\* are terminated by a ‘\0’ character so the string “Hello\0World” will be cut down to “Hello”.

28. Which of the following is used to terminate the function declaration in C++?  
a) ;  
b) ]  
c) )  
d) :  
Answer: a  
Explanation: ; semicolon is used to terminate a function declaration statement in C++.

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

1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5. char c = 74;
6. cout << c;
7. return 0;
8. }

a) I  
b) J  
c) A  
d) N  
Answer: b  
Explanation: The literal value for 74 is J. So it will be printing J.

30. What will be the output of the following C++ program?

1. #include <iomanip>
2. #include <iostream>
3. using namespace std;
4. int main()
5. {
6. cout << setprecision(17);
7. double d = 0.1;
8. cout << d << endl;
9. return 0;
10. }

a) compile time error  
b) 0.100001  
c) 0.11  
d) 0.10000000000000001  
Answer: d  
Explanation: The double had to truncate the approximation due to its limited memory, which resulted in a number that is not exactly 0.1.  
Output:

$ g++ float2.out

$ a.out

0.10000000000000001

**1. What does a Class can hold?**

A. Data

B. Functions

C. Both A and B

D. None of the above mentioned

**2. A Class is a blue print for the:**

A. Structure

B. Object

C. String

D. Character

**3. The default access level assigned to members of a class is:**

A. Private

B. Public

C. Protected

D. Needs to be assigned

**4. In which access specifier, we cannot use members outside the class:**

A. Public

B. Private

C. Protected

D. Local

**5. How many members are there in the Class?**

A. Four

B. Three

C. Two

D. One

**6. The member function of the class can be defined by:**

A. Inside the class definition

B. Outside the class definition

C. Middle of the class definition

D. Both A and B

**7. Which one of the following is the correct syntax for outside the class definition:**

A. return\_typeclass\_name::function\_name

B. return\_typeclass\_name::function\_name;

C. return\_typeclass\_name:;function\_name

D. return\_typeobject\_name::function\_name

**8. Structure is a data type in which**

A elements must have same data types

B **elements may have different data types**

C elements must have different data types

D none of these

**9. structure is a collection of**

A homogenous elements

B heterogenous elements

C homogenous elements and heterogenous elements

**10. Which statement is true in case of memory allocation of members of union**

A Memory is allocated for all variables.

B Allocates memory for variable which variable require more memory.

C Allocates memory for variable which variable require less memory.

D none of these

**11. which is true in case of union**

A require more memory space than Structure

B Declared with Struct Keyword

C require less memory space than Structure

D require more execution time than Structure

**12. C structure differs from CPP class in regards that by default all the members of the structure are \_\_\_\_\_\_\_\_\_\_ in nature.**

a. private

b. protected

c. public

d. None of these

**13. Find the output of below program**

**#include<iostream>**

**using namespace std;**

**enum color{**

(A) blue  
(B) 2  
(C) 1  
(D) None of these

**Black,**

**blue,**

**red**

**};**

**int main()**

**{ color obj = blue;**

**cout<<obj;**

**return 0;**

**}**

**14. Find the output of below program**

#include<iostream>

using namespace std;

enum color{

(A) blue  
(B) Compilation Error  
(C) 1  
(D) 2

black=1,

blue,

red };

int main()

{ color obj = blue;

cout<<obj;

return 0;

}

**15. Find Output:**

#include<iostream>

using namespace std;

(A) yellow  
(B) Compilation Error  
(C) 1  
(D) 2

enum color{

black=1,

blue,

red

};

int main()

{

color obj =yellow;

cout<<obj;

return 0;

}

1. Which rule will not affect the friend function?  
a) private and protected members of a class cannot be accessed from outside  
b) private and protected member can be accessed anywhere  
c) protected member can be accessed anywhere  
d) private member can be accessed anywhere

2. Which keyword is used to declare the friend function?  
a) firend  
b) friend  
c) classfriend  
d) myfriend

3. What is the syntax of friend function?  
a) friend class1 Class2;  
b) friend class;  
c) friend class  
d) friend class()

9. Pick out the correct statement.  
a) A friend function may be a member of another class  
b) A friend function may not be a member of another class  
c) A friend function may or may not be a member of another class  
d) None of the mentioned

. Where does keyword ‘friend’ should be placed?  
a) function declaration  
b) function definition  
c) main function  
d) block function

1. Which of the following function / types of function cannot have default parameters?

A. Member function of class  
B. Main()  
C. Member function of structure  
D. Both B and C

 Correct way to declare pure virtual function in a C++ class is

A. Virtual void foo() =0 ;  
B. Void virtual foo()= { 0 }  
C. Virtual void foo() {} = 0;  
D. None of the above

What is the scope of the variable declared in the user defined function?

A. Whole program  
B. Only inside the {} block  
C. The main function  
D. None of the above

 Which of the following statement is correct?

A. Only one parameter of a function can be a default parameter.  
B. Minimum one parameter of a function must be a default parameter.  
C. All the parameters of a function can be default parameters.  
D. No parameter of a function can be default.

1. What will be the output of the following C code?

1. #include <stdio.h>
2. int i;
3. int main()
4. {
5. **extern** int i;
6. if (i == 0)
7. printf("scope rules**\n**");
8. }

a) scope rules  
b) Compile time error due to multiple declaration  
c) Compile time error due to not defining type in statement extern i  
d) Nothing will be printed as value of i is not zero because i is an automatic variable

2. What will be the output of the following C code (without linking the source file in which ary1 is defined)?

**Get Free**[**Certificate of Merit in C Programming**](https://rank.sanfoundry.com/c-programming-certification-test/)**Now!**

1. #include <stdio.h>
2. int main()
3. {
4. **extern** ary1[];
5. printf("scope rules**\n**");
6. }

a) scope rules  
b) Linking error due to undefined reference  
c) Compile time error because size of array is not provided  
d) Compile time error because datatype of array is not provided

3. What will be the output of the following C code (after linking to source file having definition of ary1)?

1. #include <stdio.h>
2. int main()
3. {
4. **extern** ary1[];
5. printf("%d**\n**", ary1[0]);
6. }

a) Value of ary1[0];  
b) Compile time error due to multiple definition  
c) Compile time error because size of array is not provided  
d) Compile time error because datatype of array is not provided

4. What is the scope of an external variable?  
a) Whole source file in which it is defined  
b) From the point of declaration to the end of the file in which it is defined  
c) Any source file in a program  
d) From the point of declaration to the end of the file being compiled

5. What is the scope of a function?  
a) Whole source file in which it is defined  
b) From the point of declaration to the end of the file in which it is defined  
c) Any source file in a program  
d) From the point of declaration to the end of the file being compiled

1. What will be the sequence of allocation and deletion of variables in the following C code?

1. #include <stdio.h>
2. int main()
3. {
4. int a;
5. {
6. int b;
7. }
8. }

a) a->b, a->b  
b) a->b, b->a  
c) b->a, a->b  
d) b->a, b->a

2. Array sizes are optional during array declaration by using \_\_\_\_\_\_ keyword.  
a) auto  
b) static  
c) extern  
d) register

3. What will be the output of the following C code?

1. #include <stdio.h>
2. void main()
3. {
4. int x = 3;
5. {
6. x = 4;
7. printf("%d", x);
8. }
9. }

a) 4  
b) 3  
c) 0  
d) Undefined

5. What will be the output of the following C code?

1. #include <stdio.h>
2. int x;
3. void main()
4. {
5. m();
6. printf("%d", x);
7. }
8. void m()
9. {
10. x = 4;
11. }

a) 0  
b) 4  
c) Compile time error  
d) Undefined

1. What are default arguments?  
a) Arguments which are not mandatory to be passed  
b) Arguments with default value that aren’t mandatory to be passed  
c) Arguments which are not passed to functions  
d) Arguments which always take same data value

2. Which is the correct condition for the default arguments?  
a) Those must be declared as last arguments in argument list  
b) Those must be declared first in the argument list  
c) Those can be defined anywhere in the argument list  
d) Those are declared inside the function definition

3. If a member function have to be made both zero argument and parameterized constructor, which among the following can be the best option?  
a) Two normal and one default argument  
b) At least one default argument  
c) Exactly one default argument  
d) Make all the arguments default

4. Which among the following function can be called without arguments?  
a) void add(int x, int y=0)  
b) void add(int=0)  
c) void add(int x=0, int y=0)  
d) void add(char c)

5. If a function have all the default arguments but still some values are passed to the function then \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
a) The function will use the values passed to it  
b) The function will use the default values as those are local  
c) The function can use any value whichever is higher  
d) The function will choose the minimum values

6. Which among the following is correct?  
a) void test(int x=0, int y, int z=0)  
b) void test(int x=0, int=0)  
c) void test(int x, int y=0)  
d) void test(int x=’c, int y)

7. What function will be called with the independent syntax “test(5,6,7);”?  
a) void test(int x, int y)  
b) void test(int x=0, int y, int z)  
c) int test(int x=0, y=0, z=0)  
d) void test(int x, int y, int z=0)

8. Which among the following is a wrong call to the function void test(int x, int y=0, int z=0)?  
a) test(5,6,7);  
b) test(5);  
c) test();  
d) test(5,6);

9. Default arguments are \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
a) Only allowed in the parameter list of the function declaration  
b) Only allowed in the return type of the function declaration  
c) Only allowed with the class name definition  
d) Only allowed with the integer type values

10. Which among the following is false for default arguments?  
a) Those are not allowed with a declaration of pointer to functions  
b) Those are not allowed with the reference to functions  
c) Those are not allowed with the typedef declarations  
d) Those are allowed with pointer and reference to function declaration

11. The non-template functions can be added with default arguments to already declared functions \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
a) If and only if the function is declared again in the same scope  
b) If and only if the function is declared only once in the same scope  
c) If and only if the function is declared in different scope  
d) If and only if the function is declared twice in the program

12. The using declaration \_\_\_\_\_\_\_\_\_\_  
a) Doesn’t carry over the default values  
b) Carries over the known default arguments  
c) Carries over only the normal arguments  
d) Carries over only few default arguments

13. The names given to the default arguments are only looked up and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and are bound during declaration.  
a) Checked for availability  
b) Checked for random access  
c) Checked for accessibility  
d) Checked for feasibility

14. The default argument get bound during declaration \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
a) And are never executed  
b) And are executed simultaneously  
c) But are executed only if priority is given  
d) But are executed during function call

15. The virtual function overrides \_\_\_\_\_\_\_\_\_\_\_\_  
a) Do not acquire base class declaration of default arguments  
b) Do acquire base class declaration of default arguments  
c) Do not link with the default arguments of base class  
d) Do link with the default argument but only of derived classes

1. Which of the following permits function overloading on c++?  
a) type  
b) number of arguments  
c) type & number of arguments  
d) number of objects

2. In which of the following we cannot overload the function?  
a) return function  
b) caller  
c) called function  
d) main function

3. Function overloading is also similar to which of the following?  
a) operator overloading  
b) constructor overloading  
c) destructor overloading  
d) function overloading

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

1. #include <iostream>
2. using namespace std;
3. void print(int i)
4. {
5. cout << i;
6. }
7. void print(double f)
8. {
9. cout << f;
10. }
11. int main(void)
12. {
13. print(5);
14. print(500.263);
15. return 0;
16. }

a) 5500.263  
b) 500.2635  
c) 500.263  
d) 500.266

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

1. #include <iostream>
2. using namespace std;
3. int Add(int X, int Y, int Z)
4. {
5. return X + Y;
6. }
7. double Add(double X, double Y, double Z)
8. {
9. return X + Y;
10. }
11. int main()
12. {
13. cout << Add(5, 6);
14. cout << Add(5.5, 6.6);
15. return 0;
16. }

a) 11 12.1  
b) 12.1 11  
c) 11 12  
d) compile time error

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

1. #include <iostream>
2. using namespace std;
3. int operate (int a, int b)
4. {
5. return (a \* b);
6. }
7. float operate (float a, float b)
8. {
9. return (a / b);
10. }
11. int main()
12. {
13. int x = 5, y = 2;
14. float n = 5.0, m = 2.0;
15. cout << operate(x, y) <<"**\t**";
16. cout << operate (n, m);
17. return 0;
18. }

a) 10.0 5.0  
b) 5.0 2.5  
c) 10.0 5  
d) 10 2.5

7. Overloaded functions are \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
a) Very long functions that can hardly run  
b) One function containing another one or more functions inside it  
c) Two or more functions with the same name but different number of parameters or type  
d) Very long functions

8. What will happen while using pass by reference?  
a) The values of those variables are passed to the function so that it can manipulate them  
b) The location of variable in memory is passed to the function so that it can use the same memory area for its processing  
c) The function declaration should contain ampersand (& in its type declaration)  
d) The function declaration should contain $

9. What should be passed in parameters when function does not require any parameters?  
a) void  
b) blank space  
c) both void & blank space  
d) tab space

10. What are the advantages of passing arguments by reference?  
a) Changes to parameter values within the function also affect the original arguments  
b) There is need to copy parameter values (i.e. less memory used)  
c) There is no need to call constructors for parameters (i.e. faster)  
d) All of the mentioned

**1. What does the following statement mean?**

int (\*fp) (char\*);

a) pointer to a pointer  
b) pointer to an array of chars  
c) pointer to function taking a char\* argument and returns an int  
d) function taking a char\* argument and returning a pointer to int

**2. The operator used for dereferencing or indirection is \_\_\_\_**

a) \*  
b) &  
c) ->  
d) –>>

**3. Choose the right option.  
string\* x, y;**

a) x is a pointer to a string, y is a string  
b) y is a pointer to a string, x is a string  
c) both x and y are pointers to string types  
d) y is a pointer to a string

**4. Which one of the following is not a possible state for a pointer.**

a) hold the address of the specific object  
b) point one past the end of an object  
c) zero  
d) point to a type

**5. Which of the following is illegal?**

a) int \*ip;  
b) string s, \*sp = 0;  
c) int i; double\* dp = &i;  
d) int \*pi = 0;

**6. What will happen in the following C++ code snippet?**

int a =100, b =200;

int \*p = &a, \*q = &b ;

p = q

a) b is assigned to a  
b) p now points to b  
c) a is assigned to b  
d) q now points to a

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

#include <iostream>

using namespace std;

int main() {

int a = 5, b = 10, c = 15;

int \*arr[] = {&a, &b, &c};

cout << arr[1];

return 0;

}

a) 5  
b) 10  
c) 15  
d) it will return some random number

**8. The correct statement for a function that takes pointer to a float, a pointer to a pointer to a char and returns a pointer to a pointer to a integer is \_\_\_\_\_\_\_\_\_\_\_\_**

a) int \*\*fun(float\*\*, char\*\*)  
b) int \*fun(float\*, char\*)  
c) int \*\*fun(float\*, char\*\*)  
d) int \*\*\*fun(\*float, \*\*char)

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

#include <iostream>

using namespace std;

int main() {

char arr[20];

int i;

for (i = 0; i < 10; i++)

\*(arr + i) = 65 + i;

\*(arr + i) = '\0';

cout << arr;

return 0;

}

a) ABCDEFGHIJ  
b) AAAAAAAAAA  
c) JJJJJJJJ  
d) AAAAAAJJJJ

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

#include <iostream>

using namespace std;

int main() {

char \*ptr;

char str[] = 'abcdefg';

ptr = str;

ptr += 5;

cout << ptr;

return 0;

}

a) fg  
b) cdef  
c) defg  
d) abcd

Which among the following is correct for a hierarchical inheritance?  
a) Two base classes can be used to be derived into one single class  
b) Two or more classes can be derived into one class  
c) One base class can be derived into other two derived classes or more  
d) One base class can be derived into only 2 classes

Which is the correct syntax of inheritance?  
a) class derived\_classname : base\_classname{ /\*define class body\*/ };  
b) class base\_classname : derived\_classname{ /\*define class body\*/ };  
c) class derived\_classname : access base\_classname{ /\*define class body\*/ };  
d) class base\_classname :access derived\_classname{ /\*define class body\*/ };

Which type of inheritance leads to diamond problem?  
a) Single level  
b) Multi-level  
c) Multiple  
d) Hierarchical

 If a base class is inherited in protected access mode then which among the following is true?  
a) Public and Protected members of base class becomes protected members of derived class  
b) Only protected members become protected members of derived class  
c) Private, Protected and Public all members of base, become private of derived class  
d) Only private members of base, become private of derived class

 Members which are not intended to be inherited are declared as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
a) Public members  
b) Protected members  
c) Private members  
d) Private or Protected members

Single inhitance

Single level inheritance supports \_\_\_\_\_\_\_\_\_\_\_\_\_ inheritance.  
a) Runtime  
b) Compile time  
c) Multiple inheritance  
d) Language independency

Which method in the code below is single level inherited?

class A

{

protected int a, b;

public: void show()

{

cout&lt;&lt;a&lt;&lt;b;

}

};

class B: public A

{

public: void disp()

{

cout&lt;&lt;a++&lt;&lt;b++;

}

};

class C: private A, public B

{

void avg()

{

cout&lt;&lt;(a+b)/2;

}

};

1. Class A  
   b) Class B  
   c) Class C  
   d) None

If single level inheritance is used and an abstract class is created with some undefined functions, can its derived class also skip some definitions?  
a) Yes, always possible  
b) Yes, possible if only one undefined function  
c) No, at least 2 undefined functions must be there  
d) No, the derived class must implement those methods

. What is the output of the following program?

class

A

{

protected: int a,b;

public: void disp()

{

cout&lt;&lt;a&lt;&lt;b;

}

};

class B:public A

{

int x,y;

};

a) Garbage value  
b) Compile time error  
c) Runtime error  
d) Runs but gives random values as output

12. What is the output of the following program?

class A

{

float sal=40000;

}

class B extends A

{

int salBonus=10000;

public static void main(String args[])

{

B p=new B();

System.out.println("B salary is:"+p.sal);

System.out.println("Bonus of B is:"+p.bonus);

}

}

a)

B salary is: 4000.0

Bonus of B is: 10000

b)

B salary is 10000

Bonus of B is: 4000.0

c) Compile time error  
d) Runtime error

15. If base class contains 2 nested classes, will it be possible to implement single level inheritance?  
a) Yes, always  
b) Yes, only if derived class also have nested classes  
c) No, it will use more than 2 classes which is wrong  
d) No, never

Multilevel inheritance

14. Does following code show multiple inheritance?

class A

{

int a;

};

class B

{

int b;

};

class C:public A, public B

{

int c;

};

class D:public C

{

int d;

};

a) Yes, class C and class D  
b) Yes, All together it’s multilevel  
c) No, 4 classes are used  
d) No, multiple inheritance is used with class A, B and C

2. If there are 5 classes, E is derived from D, D from C, C from B and B from A. Which class constructor will be called first if the object of E or D is created?  
a) A  
b) B  
c) C  
d) A and B

1. If there are 3 classes. Class C is derived from class B and B is derived from A, Which class destructor will be called at last if object of C is destroyed.  
   a) A  
   b) B  
   c) C  
   d) All together

6. Multilevel inheritance allows \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ in the program.  
a) Only 7 levels of inheritance  
b) At least 7 levels of inheritance  
c) At most 16 levels of inheritance  
d) As many levels of inheritance as required

9. All the classes must have all the members declared private to implement multilevel inheritance.  
a) True  
b) False

12. If all the classes used parameterized constructors and no default constructor then \_\_\_\_\_\_\_\_\_\_\_  
a) The object of lower level classes can’t be created  
b) Object of lower level classes must call parent class constructors explicitly  
c) Object of lower level classes must define all the default constructors  
d) Only object of first class can be created, which is first parent

Hierarchical Inheritance

7. Which class uses hierarchical inheritance in following code?

class A

{

int a;

};

class B:class A

{

int b;

};

class C:class A,class B

{

int c;

};

class D:class A

{

int d;

};

a) Class A, B, C  
b) Class B, C, D  
c) Class A, C, D  
d) Class D, A, B

8. Which among the following is correct for following code?

abstract class A

{

public Int a;

public void disp();

};

class B:public A

{

public: void dis()

{

court&lt;&lt;a;

}

};

class C:private A

{

public void incr()

{

a++;

}

}

void main()

{

B b.disp();

}

a) Compile time error  
b) Runtime error  
c) Program runs and o/p is 0  
d) Program runs and o/p is garbage value

10. If one class have derived the base class privately then another class can’t derive the base class publically.  
a) True  
b) False

1. Hierarchical inheritance can be a subset of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   a) Hybrid inheritance  
   b) Multiple inheritance  
   c) Single level inheritance  
   d) Multilevel inheritance

14. Which class constructor is called first when an object of derived class is created?  
a) Base class constructor  
b) Derived class constructor  
c) Firstly created derived class constructor  
d) Last created derived class constructor

Hybrid inheritance

4. Diamond problem includes \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ hybrid inheritance.  
a) Hierarchical and Multiple  
b) Hierarchical and Hierarchical  
c) Multiple and Multilevel  
d) Single, Hierarchical and Multiple

5. If \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ inheritance is done continuously, it is similar to tree structure.  
a) Hierarchical  
b) Multiple  
c) Multilevel  
d) Hierarchical and Multiple

9. If hierarchical inheritance requires to inherit more than one class to single class, which syntax is correct? (A, B, C are class names)  
a) hierarchical class A: public B, public C  
b) multiple class A: public B, public C  
c) many class A: public B, public C  
d) class A: public B, public C

11. What is the minimum number of classes to be there in a program implementing hybrid inheritance?  
a) 2  
b) 3  
c) 4  
d) No limit

5. If \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ inheritance is done continuously, it is similar to tree structure.  
a) Hierarchical  
b) Multiple  
c) Multilevel  
d) Hierarchical and Multiple

Function Overloading :

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

1. #include <iostream>
2. using namespace std;
3. void print(int i)
4. {
5. cout << i;
6. }
7. void print(double f)
8. {
9. cout << f;
10. }
11. int main(void)
12. {
13. print(5);
14. print(500.263);
15. return 0;
16. }

a) 5500.263  
b) 500.2635  
c) 500.263  
d) 500.266

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

1. #include <iostream>
2. using namespace std;
3. int Add(int X, int Y, int Z)
4. {
5. return X + Y;
6. }
7. double Add(double X, double Y, double Z)
8. {
9. return X + Y;
10. }
11. int main()
12. {
13. cout << Add(5, 6);
14. cout << Add(5.5, 6.6);
15. return 0;
16. }

a) 11 12.1  
b) 12.1 11  
c) 11 12  
d) compile time error

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

1. #include <iostream>
2. using namespace std;
3. int operate (int a, int b)
4. {
5. return (a \* b);
6. }
7. float operate (float a, float b)
8. {
9. return (a / b);
10. }
11. int main()
12. {
13. int x = 5, y = 2;
14. float n = 5.0, m = 2.0;
15. cout << operate(x, y) <<"**\t**";
16. cout << operate (n, m);
17. return 0;
18. }

a) 10.0 5.0  
b) 5.0 2.5  
c) 10.0 5  
d) 10 2.5

9. What should be passed in parameters when function does not require any parameters?  
a) void  
b) blank space  
c) both void & blank space  
d) tab space

10. What are the advantages of passing arguments by reference?  
a) Changes to parameter values within the function also affect the original arguments  
b) There is need to copy parameter values (i.e. less memory used)  
c) There is no need to call constructors for parameters (i.e. faster)  
d) All of the mentioned

Operator overloading

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

#include <iostream>

#include <string>

using namespace std;

class complex

{

int i;

int j;

public:

complex(int a, int b)

{

i = a;

j = b;

}

complex operator+(complex c)

{

complex temp;

temp.i = this->i + c.i;

temp.j = this->j + c.j;

return temp;

}

void show(){

cout<<"Complex Number: "<<i<<" + i"<<j<<endl;

}

};

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

{

complex c1(1,2);

complex c2(3,4);

complex c3 = c1 + c2;

c3.show();

return 0;

}

a) 4 + i6  
b) 2 + i2  
c) Error  
d) Segmentation fault

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

#include <iostream>

#include <string>

using namespace std;

class complex

{

int i;

int j;

public:

complex(){}

complex(int a, int b)

{

i = a;

j = b;

}

complex operator+(complex c)

{

complex temp;

temp.i = this->i + c.i;

temp.j = this->j + c.j;

return temp;

}

void show(){

cout<<"Complex Number: "<<i<<" + i"<<j<<endl;

}

};

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

{

complex c1(1,2);

complex c2(3,4);

complex c3 = c1 + c2;

c3.show();

return 0;

}

a) Complex Number: 4 + i6  
b) Complex Number: 2 + i2  
c) Error  
d) Segmentation fault

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

#include <iostream>

#include <string>

using namespace std;

class complex

{

int i;

int j;

public:

complex(){}

complex(int a, int b)

{

i = a;

j = b;

}

complex operator+(complex c)

{

complex temp;

temp.i = this->i + c.i;

temp.j = this->j + c.j;

return temp;

}

void operator+(complex c)

{

complex temp;

temp.i = this->i + c.i;

temp.j = this->j + c.j;

temp.show\_poss();

}

void show(){

cout<<"Complex Number: "<<i<<" + i"<<j<<endl;

}

void show\_poss(){

cout<<"Your result after addition will be: "<<i<<" + i"<<j<<endl;

}

};

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

{

complex c1(1,2);

complex c2(3,4);

c1 + c2;

return 0;

}

a) Complex Number: 4 + i6  
b) Complex Number: 2 + i2  
c) Error  
d) Segmentation fault

8. Given the following C++ code. How would you define the < operator for Box class so that when boxes b1 and b2 are compared in if block the program gives correct result?

#include <iostream>

#include <string>

using namespace std;

class Box

{

int capacity;

public:

Box(){}

Box(double capacity){

this->capacity = capacity;

}

};

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

{

Box b1(10);

Box b2 = Box(14);

if(b1 < b2){

cout<<"Box 2 has large capacity.";

}

else{

cout<<"Box 1 has large capacity.";

}

return 0;

}

a)

bool operator<(Box b)

{

return this->capacity < b.capacity ? true : false;

}

b)

bool operator<(Box b)

{

return this->capacity > b.capacity ? true : false;

}

c)

bool operator<(Box b)

{

return b1 > b2 ? true : false;

}

d)

bool operator<(Box b)

{

return this < b ? true : false;

}

10. Pick the incorrect statements out of the following.  
a) Operator overloading does not disturbs the precedence of operators  
b) Arity of operators can be changed using operator overloading  
c) No new operators can be created  
d) All of the mentioned

9. Which is the correct statement about operator overloading?  
a) Only arithmetic operators can be overloaded  
b) Only non-arithmetic operators can be overloaded  
c) Precedence of operators are changed after overlaoding  
d) Associativity and precedence of operators does not change