**C++ Programming Multiple Choice Questions on Functions, Namespaces & Exceptions**

**Here is a listing of C++ language interview questions on “Operators” along with answers, explanations and/or solutions:**

1. Which operator is having the right to left associativity in the following?  
a) Array subscripting  
b) Function call  
c) Addition and subtraction  
d) Type cast  
View Answer

Answer: d  
Explanation: There are many rights to left associativity operators in C++, which means they are evaluation is done from right to left. Type Cast is one of them. Here is a link of the associativity of operators: https://github.com/MicrosoftDocs/cpp-docs/blob/master/docs/cpp/cpp-built-in-operators-precedence-and-associativity.

2. Which operator is having the highest precedence?  
a) postfix  
b) unary  
c) shift  
d) equality  
View Answer

Answer: a  
Explanation: The operator which is having the highest precedence is postfix and lowest is equality.

3. What is this operator called ?:?  
a) conditional  
b) relational  
c) casting operator  
d) unrelational  
View Answer

Answer: a  
Explanation: In this operator, if the condition is true means, it will return the first operator, otherwise second operator.

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

#include <iostream>

using namespace std;

int main()

{

int a;

a = 5 + 3 \* 5;

cout << a;

return 0;

}

a) 35  
b) 20  
c) 25  
d) 30  
View Answer

Answer: b  
Explanation: Because the \* operator is having highest precedence, So it is executed first and then the + operator will be executed.  
Output:

$ g++ op1.cpp

$ a.out

20

5. What is the use of dynamic\_cast operator?  
a) it converts virtual base class to derived class  
b) it converts the virtual base object to derived objects  
c) it will convert the operator based on precedence  
d) it converts the virtual base object to derived class  
View Answer

Answer: a  
Explanation: Because the dynamic\_cast operator is used to convert from base class to derived class.

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

#include <iostream>

using namespace std;

int main()

{

int a = 5, b = 6, c, d;

c = a, b;

d = (a, b);

cout << c << ' ' << d;

return 0;

}

a) 5  6  
b) 6  5  
c) 6  7  
d) 6  8  
View Answer

Answer: a  
Explanation: It is a separator here. In C, the value a is stored in c and in d the value b is stored in d because of the bracket.  
Output:

$ g++ op3.cpp

$ a.out

5 6

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

#include <iostream>

using namespace std;

int main()

{

int i, j;

j = 10;

i = (j++, j + 100, 999 + j);

cout << i;

return 0;

}

a) 1000  
b) 11  
c) 1010  
d) 1001  
View Answer

Answer: c  
Explanation: j starts with the value 10. j is then incremented to 11. Next, j is added to 100. Finally, j (still containing 11) is added to 999 which yields the result 1010.  
Output:

$ g++ op2.cpp

$ a.out

1010

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

#include <iostream>

using namespace std;

int main ()

{

int x, y;

x = 5;

y = ++x \* ++x;

cout << x << y;

x = 5;

y = x++ \* ++x;

cout << x << y;

return 0;

}

a) 749735  
b) 736749  
c) 367497  
d) 367597  
View Answer

Answer: a  
Explanation: Because of the precedence the pre-increment and post increment operator, we got the output as 749736.  
Output:

$ g++ op.cpp

$ a.out

749735

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

#include <iostream>

using namespace std;

int main()

{

int a = 5, b = 6, c;

c = (a > b) ? a : b;

cout << c;

return 0;

}

a) 6  
b) 5  
c) 4  
d) 7  
View Answer

Answer: a  
Explanation: Here the condition is false on conditional operator, so the b value is assigned to c.  
Output:

$ g++ op1.cpp

$ a.out

6

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

#include <iostream>

using namespace std;

main()

{

double a = 21.09399;

float b = 10.20;

int c ,d;

c = (int) a;

d = (int) b;

cout << c <<' '<< d;

return 0;

}

a) 20 10  
b) 10 21  
c) 21 10  
d) 10 20  
View Answer

Answer: c  
Explanation: In this program, we are casting the operator to integer, So it is printing as 21 and 10.  
Output:

$ g++ op5.cpp

$ a.out

21 10

**Here is a listing of C++ questions on “Statements” along with answers, explanations and/or solutions:**

1. How are many sequences of statements present in c++?  
a) 4  
b) 3  
c) 5  
d) 6  
View Answer

Answer: c  
Explanation: There are five sequences of statements. They are Preprocessor directives, Comments, Declarations, Function Declarations, Executable statements.

2. The if..else statement can be replaced by which operator?  
a) Bitwise operator  
b) Conditional operator  
c) Multiplicative operator  
d) Addition operator  
View Answer

Answer: b  
Explanation: In the conditional operator, it will predicate the output using the given condition.

3. The switch statement is also called as?  
a) choosing structure  
b) selective structure  
c) certain structure  
d) bitwise structure  
View Answer

Answer: b  
Explanation: The switch statement is used to choose the certain code to execute, So it is also called as selective structure.

4. The destination statement for the goto label is identified by what label?  
a) $  
b) @  
c) \*  
d) :  
View Answer

Answer: d  
Explanation: : colon is used at the end of labels of goto statements.

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

#include <iostream>

using namespace std;

int main ()

{

int n;

for (n = 5; n > 0; n--)

{

cout << n;

if (n == 3)

break;

}

return 0;

}

a) 543  
b) 54  
c) 5432  
d) 53  
View Answer

Answer: a  
Explanation: In this program, We are printing the numbers in reverse order but by using break statement we stopped printing on 3.  
Output:

$ g++ stat.cpp

$ a.out

543

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

#include <iostream>

using namespace std;

int main()

{

int a = 10;

if (a < 15)

{

time:

cout << a;

goto time;

}

break;

return 0;

}

a) 1010  
b) 10  
c) infinitely print 10  
d) compile time error  
View Answer

Answer: d  
Explanation: Because the break statement need to be presented inside a loop or a switch statement.

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

1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5. int n = 15;
6. for ( ; ;)
7. cout << n;
8. return 0;
9. }

a) error  
b) 15  
c) infinite times of printing n  
d) none of the mentioned  
View Answer

Answer: c  
Explanation: There is not a condition in the for loop, So it will loop continuously.

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

1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5. int i;
6. for (i = 0; i < 10; i++);
7. {
8. cout << i;
9. }
10. return 0;
11. }

a) 0123456789  
b) 10  
c) 012345678910  
d) compile time error  
View Answer

Answer: b  
Explanation: for loop with a semicolon is called as body less for loop. It is used only for incrementing the variable values. So in this program the value is incremented and printed as 10.  
Output:

$ g++ stat2.cpp

$ a.out

10

9. How many types of loops are there in C++?  
a) 4  
b) 2  
c) 3  
d) 1  
View Answer

Answer: c  
Explanation: There are 3 types of loop. They are “while loop”, “do while loop”, and, “for loop”. C++11 has introduced “for-each loop” or “range-based for loop”, but it’s ultimately a “for loop” only.

10. Which looping process is best used when the number of iterations is known?  
a) for  
b) while  
c) do-while  
d) all looping processes require that the iterations be known  
View Answer

Answer: a  
Explanation: Because in for loop we are allowed to provide starting and ending conditions of loops, hence fixing the number of iterations of loops, whereas no such things are provided by other loops.

**Here is a listing of C++ interview questions on “Comments and Indentation” along with answers, explanations and/or solutions:**

1. How many types of comments are there in c++?  
a) 1  
b) 2  
c) 3  
d) 4  
View Answer

Answer: b  
Explanation: There are two types of comments in C++. Single line comments uses double slash //. Multiple line comments uses /\* comment inside \*/.

2. What is a comment in c++?  
a) comments are parts of the source code disregarded by the compiler  
b) comments are executed by the compiler to find the meaning of the comment  
c) comments are executable  
d) comments are executed by the compiler  
View Answer

Answer: a  
Explanation: Comments are used to add meaning to the program.

3. What type of comments does c++ support?  
a) single line  
b) multiline  
c) single line and multi-line  
d) reusable line  
View Answer

Answer: c  
Explanation: C++ provides two types of comments in programs. They are single line(using //) or multiple line (using /\*…… \*/) comments.

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

#include <iostream>

using namespace std;

int main()

{

*/\* this is comment\**

*cout << "hello world";*

*return 0;*

*}*

a) hello world  
b) hello  
c) compile time error  
d) hellohello  
View Answer

Answer: c  
Explanation: Because the slash should need to be forward not backward.

5. What is used to write multi line comment in c++?  
a) /\* …. \*/  
b) /$ …. $/  
c) //  
d) /$ …. \*/  
View Answer

Answer: a  
Explanation: The /\* is used to write the multi line comment.

6. What is the use of the indentation in c++?  
a) distinguishes between comments and code  
b) r distinguishes between comments and outer data  
c) distinguishes between comments and outer data  
d) r distinguishes between comments and inner data  
View Answer

Answer: a  
Explanation: To distinguish between different parts of the program like comments, codes, etc.

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

#include <iostream>

using namespace std;

long factorial (long a)

{

if (a > 1)

return (a \* factorial (a + 1));

else

return (1);

}

int main ()

{

long num = 3;

cout << num << "! = " << factorial ( num );

return 0;

}

a) 6  
b) 24  
c) segmentation fault  
d) compile time error  
View Answer

Answer: c  
Explanation: As we have given in the function as a+1, it will exceed the size and so it arises the segmentation fault.  
Output:

$ g++ arg3.cpp

$ a.out

segmentation fault

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

#include <iostream>

using namespace std;

void square (int \*x)

{

\*x = (\*x + 1) \* (\*x);

}

int main ( )

{

int num = 10;

square(&num);

cout << num;

return 0;

}

a) 100  
b) compile time error  
c) 144  
d) 110  
View Answer

Answer: d  
Explanation: We have increased the x value in operand as x + 1, so it will return as 110.  
Output:

$ g++ arg2.cpp

$ a.out

110

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

#include <iostream>

using namespace std;

int add(int a, int b);

int main()

{

int i = 5, j = 6;

cout << add(i, j) << endl;

return 0;

}

int add(int a, int b )

{

int sum = a + b;

a = 7;

return a + b;

}

a) 11  
b) 12  
c) 13  
d) compile time error  
View Answer

Answer: c  
Explanation: The value of a has been changed to 7, So it returns as 13.  
Output:

$ g++ arg1.cpp

$ a.out

13

10. What will happen when we use void in argument passing?  
a) It will not return value to its caller  
b) It will return value to its caller  
c) May or may not depend on the declared return type of the function, the passed arguments are different than the function return type  
d) It will return value  
View Answer

Answer: a  
Explanation: As void is not having any return value, it will not return the value to the caller.

**Here is a listing of tough C++ programming questions on “Function Declarations” along with answers, explanations and/or solutions:**

1. Where does the execution of the program starts?  
a) user-defined function  
b) main function  
c) void function  
d) else function  
View Answer

Answer: b  
Explanation: Normally the execution of the program in c++ starts from main only.

2. What are mandatory parts in the function declaration?  
a) return type, function name  
b) return type, function name, parameters  
c) parameters, function name  
d) parameters, variables  
View Answer

Answer: a  
Explanation: In a function, return type and function name are mandatory all else are just used as a choice.

3. which of the following is used to terminate the function declaration?  
a) :  
b) )  
c) ;  
d) ]  
View Answer

Answer: c  
Explanation: ; semicolon is used to terminate a function declaration statement in C++.

4. How many can max number of arguments present in function in the c99 compiler?  
a) 99  
b) 90  
c) 102  
d) 127  
View Answer

Answer: d  
Explanation: C99 allows to pass a maximum of 127 arguments in a function.

5. Which is more effective while calling the functions?  
a) call by value  
b) call by reference  
c) call by pointer  
d) call by object  
View Answer

Answer: b  
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.

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

#include <iostream>

using namespace std;

void mani()

void mani()

{

cout<<"hai";

}

int main()

{

mani();

return 0;

}

a) hai  
b) haihai  
c) compile time error  
d) runtime error  
View Answer

Answer: c  
Explanation: We have to use the semicolon to declare the function in line 3. This is called a function declaration and a function declaration ends with a semicolon.

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

#include <iostream>

using namespace std;

void fun(int x, int y)

{

x = 20;

y = 10;

}

int main()

{

int x = 10;

fun(x, x);

cout << x;

return 0;

}

a) 10  
b) 20  
c) compile time error  
d) 30  
View Answer

Answer: a  
Explanation: In this program, we called by value so the value will not be changed, So the output is 10  
Output:

$ g++ fun.cpp

$ a.out

10

8. 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) header section  
View Answer

Answer: b  
Explanation: The variable is valid only in the function block as in other.

9. How many minimum number of functions should be present in a C++ program for its execution?  
a) 0  
b) 1  
c) 2  
d) 3  
View Answer

Answer: b  
Explanation: The execution of a C++ program starts from main function hence we require atleast 1 function to be present in a C++ program to execute and i.e. the main function.

**This set of C++ Programming Multiple Choice Questions & Answers (MCQs) focuses on “Functions”**

1. Which of the following is the default return value of functions in C++?  
a) int  
b) char  
c) float  
d) void  
View Answer

Answer: a  
Explanation: C++ uses int as the default return values for functions. It also restricts that the return type of the main function must be int.

2. What happens to a function defined inside a class without any complex operations (like looping, a large number of lines, etc)?  
a) It becomes a virtual function of the class  
b) It becomes a default calling function of the class  
c) It becomes an inline function of the class  
d) The program gives an error  
View Answer

Answer: c  
Explanation: Any function which is defined inside a class and has no complex operations like loops, a large number of lines then it is made inline.

3. What is an inline function?  
a) A function that is expanded at each call during execution  
b) A function that is called during compile time  
c) A function that is not checked for syntax errors  
d) A function that is not checked for semantic analysis  
View Answer

Answer: a  
Explanation: Inline function is those which are expanded at each call during the execution of the program to reduce the cost of jumping during execution.

4. An inline function is expanded during \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
a) compile-time  
b) run-time  
c) never expanded  
d) end of the program  
View Answer

Answer: a  
Explanation: An inline function is expanded during the compile-time of a program.

5. In which of the following cases inline functions may not word?  
i) If the function has static variables.  
ii) If the function has global and register variables.  
iii) If the function contains loops  
iv) If the function is recursive  
a) i, iv  
b) iii, iv  
c) ii, iii, iv  
d) i, iii, iv  
View Answer

Answer: d  
Explanation: A function is not inline if it has static variables, loops or the function is having any recursive calls.

6. When we define the default values for a function?  
a) When a function is defined  
b) When a function is declared  
c) When the scope of the function is over  
d) When a function is called  
View Answer

Answer: b  
Explanation: Default values for a function is defined when the function is declared inside a program.

7. Where should default parameters appear in a function prototype?  
a) To the rightmost side of the parameter list  
b) To the leftmost side of the parameter list  
c) Anywhere inside the parameter list  
d) Middle of the parameter list  
View Answer

Answer: a  
Explanation: Default parameters are defined to the rightmost side of parameter list in a function to differentiate between the normal and default parameters for example if a function is defined as fun(int x = 5, int y) then if we call fun(10) then 10 should be given to x or y because one can apply both logics like x = 10 already defined and 10 passed is for y but if compiler reads it from left to right it will think it is for x and no parameter is given for y, therefore, the compiler will give error.

8. If an argument from the parameter list of a function is defined constant then \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
a) It can be modified inside the function  
b) It cannot be modified inside the function  
c) Error occurs  
d) Segmentation fault  
View Answer

Answer: b  
Explanation: A function is not allowed a constant member of the parameter list.

9. Which of the following feature is used in function overloading and function with default argument?  
a) Encapsulation  
b) Polymorphism  
c) Abstraction  
d) Modularity  
View Answer

Answer: b  
Explanation: Both of the above types allows a function overloading which is the basic concept of Polymorphism.

**Here is a listing of tough C++ programming questions on “Argument Passing” along with answers, explanations and/or solutions:**

1. How many ways of passing a parameter are there in c++?  
a) 1  
b) 2  
c) 3  
d) 4  
View Answer

Answer: c  
Explanation: There are three ways of passing a parameter. They are pass by value,pass by reference and pass by pointer.

2. Which is used to keep the call by reference value as intact?  
a) static  
b) const  
c) absolute  
d) virtual  
View Answer

Answer: b  
Explanation: Because const will not change the value of the variables during the execution.

3. By default how the value are passed in c++?  
a) call by value  
b) call by reference  
c) call by pointer  
d) call by object  
View Answer

Answer: a  
Explanation: None.

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

#include <iostream>

using namespace std;

void copy (int& a, int& b, int& c)

{

a \*= 2;

b \*= 2;

c \*= 2;

}

int main ()

{

int x = 1, y = 3, z = 7;

copy (x, y, z);

cout << "x =" << x << ", y =" << y << ", z =" << z;

return 0;

}

a) 2 5 10  
b) 2 4 5  
c) 2 6 14  
d) 2 4 9  
View Answer

Answer: c  
Explanation: Because we multiplied the values by 2 in the copy function.

Output:

$ g++ arg6.cpp

$ a.out

x = 2,y = 6,z = 14

5. What will be the new value of x in the following C++ code?

#include <iostream>

using namespace std;

void fun(int &x)

{

x = 20;

}

int main()

{

int x = 10;

fun(x);

cout << "New value of x is " << x;

return 0;

}

a) 10  
b) 20  
c) 15  
d) 36  
View Answer

Answer: b  
Explanation: As the parameter is passed by reference, the value in the original memory of x is changed hence the output is printed as 20.  
Output:

$ g++ arg5.cpp

$ a.out

20

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

#include <iostream>

using namespace std;

long factorial (long a)

{

if (a > 1)

return (a \* factorial (a + 1));

else

return (1);

}

int main ()

{

long num = 3;

cout << num << "! = " << factorial ( num );

return 0;

}

a) 6  
b) 24  
c) segmentation fault  
d) compile time error  
View Answer

Answer: c  
Explanation: As we have given in the function as a+1, it will exceed the size and so it arises the segmentation fault.  
Output:

$ g++ arg3.cpp

$ a.out

segmentation fault

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

#include <iostream>

using namespace std;

void square (int \*x)

{

\*x = (\*x + 1) \* (\*x);

}

int main ( )

{

int num = 10;

square(&num);

cout << num;

return 0;

}

a) 100  
b) compile time error  
c) 144  
d) 110  
View Answer

Answer: d  
Explanation: We have increased the x value in operand as x+1, so it will return as 110.  
Output:

$ g++ arg2.cpp

$ a.out

110

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

#include <iostream>

using namespace std;

int add(int a, int b);

int main()

{

int i = 5, j = 6;

cout << add(i, j) << endl;

return 0;

}

int add(int a, int b )

{

int sum = a + b;

a = 7;

return a + b;

}

a) 11  
b) 12  
c) 13  
d) compile time error  
View Answer

Answer: c  
Explanation: The value of a has been changed to 7, So it returns as 13.  
Output:

$ g++ arg1.cpp

$ a.out

13

9. What will happen when we use void in argument passing?  
a) It will not return value to its caller  
b) It will return value to its caller  
c) Maybe or may not be return any value to its caller  
d) It will return value with help of object  
View Answer

Answer: a  
Explanation: As void is not having any return value, it will not return the value to the caller.

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

#include <iostream>

using namespace std;

void Sum(int a, int b, int & c)

{

a = b + c;

b = a + c;

c = a + b;

}

int main()

{

int x = 2, y =3;

Sum(x, y, y);

cout << x << " " << y;

return 0;

}

1. 2 3
2. 6 9
3. 2 15
4. compile time error

View Answer

Answer: c  
Explanation: We have passed three values and it will manipulate according to the given condition and yield the result as 2 15  
Output:

$ g++ arg.cpp

$ a.out

2 15

**Here is a listing of C++ interview questions on “Value Return” along with answers, explanations and/or solutions:**

1. How many types of returning values are present in c++?  
a) 1  
b) 2  
c) 3  
d) 4  
View Answer

Answer: c  
Explanation: The three types of returning values are return by value, return by reference and return by address.

2. What will you use if you are not intended to get a return value?  
a) static  
b) const  
c) volatile  
d) void  
View Answer

Answer: d  
Explanation: Void is used to not to return anything.

3. Where does the return statement returns the execution of the program?  
a) main function  
b) caller function  
c) same function  
d) block function  
View Answer

Answer: b  
Explanation: The execution of the program is returned to the point from where the function was called and the function from which this function was called is known as caller function.

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

#include <iostream>

using namespace std;

int max(int a, int b )

{

return ( a > b ? a : b );

}

int main()

{

int i = 5;

int j = 7;

cout << max(i, j );

return 0;

}

a) 5  
b) 7  
c) either 5 or 7  
d) 13  
View Answer

Answer: b  
Explanation: In this program, we are returning the maximum value by using conditional operator.  
Output:

$ g++ ret.cpp

$ a.out

7

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

#include <iostream>

using namespace std;

double & WeeklyHours()

{

double h = 46.50;

double &hours = h;

return hours;

}

int main()

{

double hours = WeeklyHours();

cout << "Weekly Hours: " << hours;

return 0;

}

a) 46.5  
b) 6.50  
c) compile time error  
d) 26.5  
View Answer

Answer: a  
Explanation: We are returning the value what we get as input.  
Output:

$ g++ ret1.cpp

$ a.out

46.5

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

#include <iostream>

using namespace std;

int mult (int x, int y)

{

int result;

result = 0;

while (y != 0)

{

result = result + x;

y = y - 1;

}

return(result);

}

int main ()

{

int x = 5, y = 5;

cout << mult(x, y) ;

return(0);

}

a) 20  
b) 25  
c) 30  
d) 35  
View Answer

Answer: b  
Explanation: We are multiplying these values by adding every values.  
Output:

$ g++ ret.cpp

$ a.out

25

7. When will we use the function overloading?  
a) same function name but different number of arguments  
b) different function name but same number of arguments  
c) same function name but same number of arguments  
d) different function name but different number of arguments  
View Answer

Answer: a  
Explanation: We use function overloading when we want the same name function to perform different procedure for different types of parameters or different number of parameters provided to the function.

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

#include <iostream>

using namespace std;

int gcd (int a, int b)

{

int temp;

while (b != 0)

{

temp = a % b;

a = b;

b = temp;

}

return(a);

}

int main ()

{

int x = 15, y = 25;

cout << gcd(x, y);

return(0);

}

a) 15  
b) 25  
c) 375  
d) 5  
View Answer

Answer: d  
Explanation: In this program, we are finding the gcd of the number.  
Output:

$ g++ ret5.cpp

$ a.out

5

**Here is a listing of C++ language interview questions on “Overloaded Function Names” along with answers, explanations and/or solutions:**

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  
View Answer

Answer: c  
Explanation: Both type and number of arguments permits function overloading in C++, like  
int func(int);  
float func(float, float)  
Here both type and number of arguments are different.

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

View Answer

Answer: a  
Explanation: While overloading the return function, it will rise a error, So we can’t overload the return function.

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

Answer: b  
Explanation: In constructor overloading, we will be using the same options availed in function overloading.

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

#include <iostream>

using namespace std;

void print(int i)

{

cout << i;

}

void print(double f)

{

cout << f;

}

int main(void)

{

print(5);

print(500.263);

return 0;

}

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

Answer: a  
Explanation: In this program, we are printing the values and the values will be print(5) will be printed first because of the order of the execution.  
Output:

**Note: Join free Sanfoundry classes at**[**Telegram**](https://t.me/sanfoundryclasses/)**or [Youtube](https://www.youtube.com/c/SanfoundryOfficial/?sub_confirmation=1" \t "_blank)**

$ g++ over.cpp

$ a.out

5500.263

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

#include <iostream>

using namespace std;

int Add(int X, int Y, int Z)

{

return X + Y;

}

double Add(double X, double Y, double Z)

{

return X + Y;

}

int main()

{

cout << Add(5, 6);

cout << Add(5.5, 6.6);

return 0;

}

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

Answer: d  
Explanation: As one can observe that no function has declaration similar to that of called Add(int, int) and Add(double, double) functions. Therefore, error occurs.

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

#include <iostream>

using namespace std;

int operate (int a, int b)

{

return (a \* b);

}

float operate (float a, float b)

{

return (a / b);

}

int main()

{

int x = 5, y = 2;

float n = 5.0, m = 2.0;

cout << operate(x, y) <<"**\t**";

cout << operate (n, m);

return 0;

}

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

Answer: d  
Explanation: In this program, we are divide and multiply the values.  
Output:

$ g++ over3.cpp

$ a.out

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  
View Answer

Answer: c  
Explanation: This is the definition of function overloading i.e. function having same name but different number of parameters and types.

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 $  
View Answer

Answer: b  
Explanation: In pass by reference, we can use the function to access the variable and it can modify it. Therefore we are using pass by reference.

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  
View Answer

Answer: b  
Explanation: When we does not want to pass any argument to a function then we leave the parameters blank i.e. func() – function without any parameter.

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  
View Answer

Answer: d  
Explanation: All the above mentioned are advantages and properties of call by reference.

**Here is a listing of C++ questions and puzzles on “Default Arguments” along with answers, explanations and/or solutions:**

1. If the user did not supply the value, what value will it take?  
a) default value  
b) rise an error  
c) both default value & rise an error  
d) error  
View Answer

Answer: a  
Explanation: If the user did not supply the value means, the compiler will take the given value in the argument list.

2. Where can the default parameter be placed by the user?  
a) leftmost  
b) rightmost  
c) both leftmost & rightmost  
d) topmost  
View Answer

Answer: b  
Explanation: To avoid the ambiguity between the non-default parameters and default parameters.

3. Which value will it take when both user and default values are given?  
a) user value  
b) default value  
c) custom value  
d) defined value  
View Answer

Answer: a  
Explanation: The default value will be used when the user value is not given, So in this case, the user value will be taken.

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

#include <iostream>

using namespace std;

void func(int a, bool flag = true)

{

if (flag == true )

{

cout << "Flag is true. a = " << a;

}

else

{

cout << "Flag is false. a = " << a;

}

}

int main()

{

func(200, false);

return 0;

}

a) Flag is true. a = 200  
b) Flag is false. a = 100  
c) Flag is false. a = 200  
d) Flag is true. a = 100  
View Answer

Answer: c  
Explanation: In this program, we are passing the value, as it evaluates to false, it produces the output as following.  
Output:

$ g++ def.cpp

$ a.out

Flag is false. a = 200

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

#include <iostream>

#include <string>

using namespace std;

string askNumber(string prompt = "Please enter a number: ");

int main()

{

string number = askNumber();

cout << "Here is your number: " << number;

return 0;

}

string askNumber(string prompt)

{

string number;

cout << prompt;

cin >> number;

return number;

}

1. 5  
   b) 6  
   c) the number you entered  
   d) compile time error  
   View Answer

Answer: c  
Explanation: In this program, we are getting a number and printing it.  
Output:

$ g++ def1.cpp

$ a.out

Please enter a number:

5

Here is your number:5

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

#include <iostream>

using namespace std;

void Values(int n1, int n2 = 10)

{

using namespace std;

cout << "1st value: " << n1;

cout << "2nd value: " << n2;

}

int main()

{

Values(1);

Values(3, 4);

return 0;

}

a)

1st value: 1

10

3

4

b)

1st value: 1

10

3

10

c) compile time error  
d) runtime error  
View Answer

Answer: a  
Explanation: In this program, We are passing the values as by default values rules it is working.  
Output:

$ g++ def2.cpp

$ a.out

1st value: 1

2nd value: 10

1st value: 3

2nd value: 4

7. What we can’t place followed by the non-default arguments?  
a) trailing arguments  
b) default arguments  
c) both trailing & default arguments  
d) leading arguments  
View Answer

Answer: b  
Explanation: To avoid the ambiguity in arguments.  
eg. if func(int a=3, int b);  
so if we call func(5), here will 5 will be value of a or b, because 5 is first parameter so a should be 5 but as only one argument is given b should be 5. So to remove such ambiguity default parameters are kept at the end or rightmost side.

8. If we start our function call with default arguments means, what will be proceeding arguments?  
a) user argument  
b) empty arguments  
c) default arguments  
d) user & empty arguments  
View Answer

Answer: c  
Explanation: As a rule, the default argument must be followed by default arguments only.

9. What is the default return type of a function?  
a) int  
b) void  
c) float  
d) char  
View Answer

Answer: b  
Explanation: void is the default return value of any function, to handle both empty and non-empty values.

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

#include <iostream>

using namespace std;

int func(int m = 10, int n)

{

int c;

c = m + n;

return c;

}

int main()

{

cout << func(5);

return 0;

}

a) 15  
b) 10  
c) compile time error  
d) 30  
View Answer

Answer: c  
Explanation: In function parameters the default arguments should always be the rightmost parameters.

**Here is a listing of C++ questions and puzzles on “Pointer to Function” along with answers, explanations and/or solutions:**

1. To which does the function pointer point to?  
a) variable  
b) constants  
c) function  
d) absolute variables  
View Answer

Answer: c  
Explanation: A function pointer points to a function.

2. What will we not do with function pointers?  
a) allocation of memory  
b) deallocation of memory  
c) both allocation & deallocation of memory  
d) finds memory status  
View Answer

Answer: c  
Explanation: As it is used to execute a block of code, So we will not allocate or deallocate memory.

3. What is the default calling convention for a compiler in c++?  
a) \_\_cdecl  
b) \_\_stdcall  
c) \_\_pascal  
d) \_\_fastcall  
View Answer

Answer: a  
Explanation: \_\_cdecl is the default calling convention for a compiler in c++.

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

#include <iostream>

using namespace std;

int add(int first, int second)

{

return first + second + 15;

}

int operation(int first, int second, int (\*functocall)(int, int))

{

return (\*functocall)(first, second);

}

int main()

{

int a;

int (\*plus)(int, int) = add;

a = operation(15, 10, plus);

cout << a;

return 0;

}

a) 25  
b) 35  
c) 40  
d) 45  
View Answer

Answer: c  
Explanation: In this program, we are adding two numbers with 15, So we got the output as 40.  
Output:

$ g++ pfu2.cpp

$ a.out

40

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

#include <iostream>

using namespace std;

void func(int x)

{

cout << x ;

}

int main()

{

void (\*n)(int);

n = &func;

(\*n)( 2 );

n( 2 );

return 0;

}

a) 2  
b) 20  
c) 21  
d) 22  
View Answer

Answer: d  
Explanation: As we are calling the function two times with the same value, So it is printing as 22.  
Output:

$ g++ pfu.cpp

$ a.out

22

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

#include <iostream>

using namespace std;

int n(char, int);

int (\*p) (char, int) = n;

int main()

{

(\*p)('d', 9);

p(10, 9);

return 0;

}

int n(char c, int i)

{

cout << c << i;

return 0;

}

a)

d9

9

b) d9d9  
c) d9  
d) compile time error  
View Answer

Answer: a  
Explanation: As function pointer p is pointing to n(char, int), so for first call d9 will be printed for second call 10, which corresponds to ‘\n’ character, and then 9 is printed.  
Output:

$ g++ pfu1.cpp

$ a.out

d9

9

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

#include <iostream>

using namespace std;

int func (int a, int b)

{

cout << a;

cout << b;

return 0;

}

int main(void)

{

int(\*ptr)(char, int);

ptr = func;

func(2, 3);

ptr(2, 3);

return 0;

}

a) 2323  
b) 232  
c) 23  
d) compile time error  
View Answer

Answer: d  
Explanation: In this program, we can’t do the casting from char to int, So it is raising an error.

8. What is the mandatory part to present in function pointers?  
a) &  
b) return values  
c) data types  
d) $  
View Answer

Answer: c  
Explanation: The data types are mandatory for declaring the variables in the function pointers.

9. which of the following can be passed in function pointers?  
a) variables  
b) data types  
c) functions  
d) objects  
View Answer

Answer: c  
Explanation: Only functions are passed in function pointers.

10. What is the meaning of the following declaration?

int(\*ptr[5])();

a) ptr is pointer to function  
b) ptr is array of pointer to function  
c) ptr is pointer to such function which return type is array  
d) ptr is pointer to array of function  
View Answer

Answer: b  
Explanation: In this expression, ptr is array not pointer.

**Here is a listing of C++ Questions & Answers focuses on “Modularization and Interfaces” along with answers, explanations and/or solutions:**

1. which of the following is used to implement the c++ interfaces?  
a) absolute variables  
b) abstract classes  
c) constant variables  
d) default variables  
View Answer

Answer: b  
Explanation: Abstract classes in C++ are purposely defined for making base classes containing atleast one virtual function which can be overloaded on inheritance, which means single function name for different sub-classes, hence acts as an interface.

2. What is the ability to group some lines of code that can be included?  
in the program?  
a) specific task  
b) program control  
c) modularization  
d) macros  
View Answer

Answer: c  
Explanation: Modularization is also similar to macros but it is used to build large projects.

3. How many types do functions fall depends on modularization?  
a) 1  
b) 2  
c) 3  
d) 4  
View Answer

Answer: b  
Explanation: There are two types of functions. They are program control and specific task.

4. How many types of modularization are there in c++?  
a) 4  
b) 3  
c) 1  
d) 2  
View Answer

Answer: d  
Explanation: There are two types of modular programming. They are interface and implementation.

5. What does the client module import?  
a) macro  
b) records  
c) interface  
d) instance  
View Answer

Answer: c  
Explanation: Because they access the functions in the module user interface.

6. Identify the correct statement.  
a) c++ does not have built-in interfaces  
b) c++ does have built-in interfaces  
c) c++ have no concept of interfaces  
d) c++ does have built-in interfaces & classes  
View Answer

Answer: a  
Explanation: Unlike other programming languages like Java and others, C++ has no inbuilt interfaces.

7. What is similar to the interface in c++?  
a) methods  
b) instance of a class  
c) pure abstract class  
d) methods & instance of a class  
View Answer

Answer: c  
Explanation: Pure abstract classes in C++ are a type of interface because it contains only abstract member functions and no data or concrete member functions.

8. Which of the following implements the module in the program?  
a) macro  
b) header files  
c) macro & header files  
d) interfaces  
View Answer

Answer: b  
Explanation: We can include the group of code by using the #include header file.

**Here is a listing of C++ questions and puzzles on “Namespaces” along with answers, explanations and/or solutions:**

1. Which operator is used to signify the namespace?  
a) conditional operator  
b) ternary operator  
c) scope operator  
d) bitwise operator  
View Answer

Answer: c  
Explanation: Scope operator(::) is used in namespace syntax.  
General syntax:  
namespace X{ int a;}  
cout<<X::a;

2. Identify the correct statement.  
a) Namespace is used to group class, objects and functions  
b) Namespace is used to mark the beginning of the program  
c) A namespace is used to separate the class, objects  
d) Namespace is used to mark the beginning & end of the program  
View Answer

Answer: a  
Explanation: Namespace allows you to group class, objects, and functions. It is used to divide the global scope into the sub-scopes.

3. What is the use of Namespace?  
a) To encapsulate the data  
b) To structure a program into logical units  
c) Encapsulate the data & structure a program into logical units  
d) It is used to mark the beginning of the program  
View Answer

Answer: b  
Explanation: The main aim of the namespace is to understand the logical units of the program and to make the program so robust.

4. What is the general syntax for accessing the namespace variable?  
a) namespace::operator  
b) namespace,operator  
c) namespace#operator  
d) namespace$operator  
View Answer

Answer: a  
Explanation: To access variables from namespace we use following syntax.  
namespace :: variable;  
General syntax:  
namespace X{ int a;}  
cout<<X::a;

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

#include <iostream>

using namespace std;

namespace first

{

int var = 5;

}

namespace second

{

double var = 3.1416;

}

int main ()

{

int a;

a = first::var + second::var;

cout << a;

return 0;

}

a) 8.31416  
b) 8  
c) 9  
d) compile time error  
View Answer

Answer: b  
Explanation: As we are getting two variables from namespace variable and we are adding that.  
Output:

$ g++ name.cpp

$ a.out

8

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

#include <iostream>

using namespace std;

namespace first

{

int x = 5;

int y = 10;

}

namespace second

{

double x = 3.1416;

double y = 2.7183;

}

int main ()

{

using first::x;

using second::y;

bool a, b;

a = x > y;

b = first::y < second::x;

cout << a << b;

return 0;

}

a) 11  
b) 01  
c) 00  
d) 10  
View Answer

Answer: d  
Explanation: We are inter mixing the variable and comparing it which is bigger and smaller and according to that we are printing the output.  
Output:

$ g++ name1.cpp

$ a.out

10

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

#include <iostream>

using namespace std;

namespace Box1

{

int a = 4;

}

namespace Box2

{

int a = 13;

}

int main ()

{

int a = 16;

Box1::a;

Box2::a;

cout << a;

return 0;

}

a) 4  
b) 13  
c) 16  
d) compile time error  
View Answer

Answer: c  
Explanation: In this program, as there is lot of variable a and it is printing the value inside the block because it got the highest priority.  
Output:

$ g++ name2.cpp

$ a.out

16

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

#include <iostream>

using namespace std;

namespace space

{

int x = 10;

}

namespace space

{

int y = 15;

}

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

{

space::x = space::y =5;

cout << space::x << space::y;

}

a) 1015  
b) 1510  
c) 55  
d) compile time error  
View Answer

Answer: c  
Explanation: We are overriding the value at the main function and so we are getting the output as 55.  
Output:

$ g++ name4.cpp

$ a.out

55

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

#include <iostream>

using namespace std;

namespace extra

{

int i;

}

void i()

{

using namespace extra;

int i;

i = 9;

cout << i;

}

int main()

{

enum letter { i, j};

class i { letter j; };

::i();

return 0;

}

a) 9  
b) 10  
c) compile time error  
d) 11  
View Answer

Answer: a  
Explanation: A scope resolution operator without a scope qualifier refers to the global namespace.

**Here is a listing of C++ Programming Questions & Answers focuses on “Exceptions” along with answers, explanations and/or solutions:**

1. To where does the program control transfers when the exception is arisen?  
a) catch  
b) handlers  
c) throw  
d) try  
View Answer

Answer: b  
Explanation: When an exception is arisen mean, the exception is caught by handlers and then it decides the type of exception.

2. Which keyword is used to check exception in the block of code?  
a) catch  
b) throw  
c) try  
d) handlers  
View Answer

Answer: c  
Explanation: The try() statement is used for exceptions in c++.

3. What will happen when the exception is not caught in the program?  
a) error  
b) program will execute  
c) block of that code will not execute  
d) program will execute & displays wrong output  
View Answer

Answer: a  
Explanation: When exceptions are not caught in any program then program throws error.

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

#include <iostream>

using namespace std;

int main()

{

int age = 0;

try

{

if (age < 0)

{

throw "Positive Number Required";

}

cout << age;

}

catch(const char \*Message)

{

cout << "Error: " << Message;

}

return 0;

}

a) 0  
b) error:Positive Number Required  
c) compile time error  
d) runtime error  
View Answer

Answer: a  
Explanation: As the zero marks the beginning of the positive number, it is printed as output  
Output:

$ g++ excep.cpp

$ a.out

0

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

#include <iostream>

using namespace std;

void PrintSequence(int StopNum)

{

int Num;

Num = 1;

while (true)

{

if (Num >= StopNum)

throw Num;

cout << Num;

Num++;

}

}

int main(void)

{

try

{

PrintSequence(20);

}

catch(int ExNum)

{

cout << "Caught an exception with value: " << ExNum;

}

return 0;

}

a) compile time error  
b) prints first 19 numbers  
c) prints first 19 numbers and throws exception at 20  
d) prints first 17 numbers  
View Answer

Answer: c  
Explanation: In this program, we are printing upto 19 numbers and when executing the 20, we are raising a exception.  
Output:

$ g++ excep1.cpp

$ a.out

12345678910111213141516171819Caught an exception with value: 20

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

#include <iostream>

using namespace std;

double division(int a, int b)

{

if (b == 0)

{

throw "Division by zero condition!";

}

return (a / b);

}

int main ()

{

int x = 50;

int y = 2;

double z = 0;

try

{

z = division(x, y);

cout << z;

}

catch(const char \*msg)

{

cerr << msg;

}

return 0;

}

a) 25  
b) 20  
c) Division by zero condition!  
d) 35  
View Answer

Answer: a  
Explanation: In this program, we resembling the division by using the exception handling.  
Output:

$ g++ excep2.cpp

$ a.out

25

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

#include <iostream>

using namespace std;

int main()

{

char\* buff;

try

{

buff = new char[1024];

if (buff == 0)

throw "Memory allocation failure!";

else

cout << sizeof(buff) << "Byte successfully allocated!"<<endl;

}

catch(char \*strg)

{

cout<<"Exception raised: "<<strg<<endl;

}

return 0;

}

a) 4 Bytes allocated successfully  
b) 8 Bytes allocated successfully  
c) Memory allocation failure  
d) Depends on the size of the data type  
View Answer

Answer: d  
Explanation: As we are allocating the memory to the variables and if there are not sufficient size means, it will throw an exception.  
Output:

$ g++ excep3.cpp

$ a.out

4 Bytes allocated successfully

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

#include <iostream>

using namespace std;

void Funct();

int main()

{

try

{

Funct();

}

catch(double)

{

cerr << "caught a double type..." << endl;

}

return 0;

}

void Funct()

{

throw 3;

}

a) caught a double type  
b) compile time error  
c) abnormal program termination  
d) runtime error  
View Answer

Answer: c  
Explanation: As we are throwing integer to double it will raise as abnormal program after termination throw statement.  
Output:

$ g++ excep4.cpp

$ a.out

terminate called after throwing an instance of 'int'

Aborted

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

#include <iostream>

#include <exception>

using namespace std;

int main()

{

try

{

int \* array1 = new int[100000000];

int \* array2 = new int[100000000];

int \* array3 = new int[100000000];

int \* array4 = new int[100000000];

cout << "Allocated successfully";

}

catch(bad\_alloc&)

{

cout << "Error allocating the requested memory." << endl;

}

return 0;

}

a) Allocated successfully  
b) Error allocating the requested memory  
c) Depends on the memory of the computer  
d) Error  
View Answer

Answer: c  
Explanation: In this program, we allocating the memory to the arrays by using exception handling and we handled the exception by standard exception.  
Output:

$ g++ excep5.cpp

$ a.out

Allocated successfully

10. What will happen when the handler is not found for an exception?  
a) calls the standard library function terminate()  
b) raise an error  
c) executes the remaining block  
d) raise an error and executes the remaining block  
View Answer

Answer: a  
Explanation: None.