**C++ BASICS**

1. C++ is \_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. procedural programming language
3. object oriented programming language
4. functional programming language
5. Both procedural and object oriented programming language.

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

1. What is the output of this program?
2. #include <iostream>
3. using namespace std;
4. int main()
5. {
6. char \*ptr;
7. char Str[] = "abcdefg";
8. ptr = Str;
9. ptr += 5;
10. cout << ptr;
11. return 0;
12. }
    1. Fg
    2. cdef
    3. defg
    4. Abcd

Answer: a  
Explanation: Pointer ptr points to string ‘fg’. So it prints fg.

1. How many objects can present in a single class?

a) 1  
b) 2  
c) 3  
d) as many as possible

Answer: d  
Explanation: Because a class may contain any number of objects according to its compliance.

1. Which is used to create a pure virtual function?

a) $  
b) =0  
c) &  
d) !

Answer: b  
Explanation: For making a method as pure virtual function, We have to append ‘=0’ to the class or method.

1. Pick out the correct statement about vector.

a) vector<int> values (5)  
b) vector values (5)  
c) vector<int> (5)  
d) none of the mentioned

Answer: a  
Explanation: The syntax for declaring the vector element is vector<type> variable\_name (number\_of\_elements);

1. Which of the following is used to implement the c++ interfaces?

a) absolute variables

b) abstract classes

c) constant variables

d) none of the mentioned

1. The default executable generation on UNIX for a C++ program is \_\_\_

[a) a.exe](javascript:void(0);)

[b) a](javascript:void(0);)

c) a.out

[d) out.a](javascript:void(0);)

Answer: c

1. How many objects can be created from an abstract class?
   * 1. 0
     2. 1
     3. 2
     4. As many as we want

Answer: a

1. What is the output of following program?

#include<iostream>

using namespace std;

int x = 10;

void fun()

{

int x = 2;

{

int x = 1;

cout << ::x << endl;

}

}

int main()

{

fun();

return 0;

}

* + 1. 2
    2. 1
    3. 10
    4. Error

Answer: c;

Explanation: If [Scope Resolution Operator](http://en.wikipedia.org/wiki/Scope_resolution_operator) is placed before a variable name then the global variable is referenced. So if we remove the following line from the above program then it will fail in compilation.

1. Which is correct with respect to size of the data types?  
   * 1. char > int < float  
        b) int < char > float  
        c) char < int < float  
        d) char < int < double

Answer: d  
Explanation: The char has less bytes than int and int has less bytes than double whereas int and float can potentially have same sizes.

1. What is the output of this program?
   1. #include <iostream>
   2. using namespace std;
   3. void mani()
   4. void mani()
   5. {
   6. cout<<"hai";
   7. }
   8. int main()
   9. {
   10. mani();
   11. return 0;
   12. }
       1. Hai
       2. Haihai
       3. compile time error
       4. none of the mentioned

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

1. Where does the return statement returns the execution of the program?  
   a) main function  
   b) caller function  
   c) same function  
   d) none of the mentioned

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.

1. What is the output of this program?

// Assume that integers take 4 bytes.

#include<iostream>

using namespace std;

class Test

{

static int i;

int j;

};

int Test::i;

int main()

{

cout << sizeof(Test);

return 0;

}

* + 1. 2
    2. 4
    3. 0
    4. Can’t predict

Answer: b;

Explanation: static data members do not contribute in size of an object. So ‘i’ is not considered in size of Test. Also, all functions (static and non-static both) do not contribute in size.

1. What is the output of this program?

#include<iostream>

using namespace std;

class Base1 {

public:

Base1()

{ cout << " Base1's constructor called" << endl; }

};

class Base2 {

public:

Base2()

{ cout << "Base2's constructor called" << endl; }

};

class Derived: public Base1, public Base2 {

public:

Derived()

{ cout << "Derived's constructor called" << endl; }

};

int main()

{

Derived d;

return 0;

}

1. Base1’s constructor called  
   Base2’s constructor called  
   Derived’s constructor called
2. Base2’s constructor called  
   Base1’s constructor called  
   Derived’s constructor called
3. Derived’s constructor called

Base1’s constructor called  
Base2’s constructor called

1. Derived’s constructor called

Base2’s constructor called  
Base1’s constructor called

Answer: a

In case of Multiple Inheritance, constructors of base classes are always called in derivation order from left to right and Destructors are called in reverse order.

1. If Integer takes 4 bytes, what is the output of this program?

#include<iostream>

using namespace std;

class base {

int arr[10];

};

class b1: public base { };

class b2: public base { };

class derived: public b1, public b2 {};

int main(void)

{

cout<<sizeof(derived);

getchar();

return 0;

}

1. 0
2. 40
3. 20
4. 80

Answer: d

Since b1 and b2 both inherit from class base, two copies of class base are there in class derived.

1. When a copy constructor may be called?
   * 1. When an object of the class is returned by value
     2. When an object of the class is passed (to a function) by value as an argument.
     3. When an object is constructed based on another object of the same class
     4. When compiler generates a temporary object.
     5. All of the above

Answer: e

1. What is the output of this program?

#include<iostream>

using namespace std;

class Point {

    Point() { cout << "Constructor called"; }

};

int main()

{

   Point t1;

   return 0;

}

1. Runtime error
2. Compile time error
3. Constructor called

Answer: b;

By default all members of a class are private. Since no access specifier is there for Point(), it becomes private and it is called outside the class when t1 is constructed in main.