

NO STL

Page 1/5

C++ Test (Pre OOP test)

19 July 2021

Pointer Size:
8 bytes

Time: 1 hr

[1] Q1.

Describe auto keyword in one line.

[3] Q2.

Assuming y is of type int, find the data types of a, b & c:

auto a = *y;

auto b = &y;

auto &b = *y;

Also, report compiler errors if any.

[2+2+2] Q3.

Find the size of following structs:

(a)

struct A

{ char c; int i; double d; char* p; };

(b)

struct B

{ int i; double* p1; char* p2; char c; };

(c)

struct C

{ double d; char c; int i; int* x; };

[2] Q4.

Create a struct D such that its size is minimum and have the following variables:

double d; char c; int i; long long l; char* p;

[1+3] Q5. (a) In above question, some space will be wasted due to ____.

(b) Write a simple C++ program that writes character 'a' at every wasted gap of the following variable:

D d;

[2+2+2] Q6. Define the following with an appropriate code:

(a) Memory leak

(c) Null pointer

(b) Dangling pointer

[0.5*8] Q7. In the following program, highlight global, function and local scope (at each line mentioned). There can be multiple scopes at a single place. Write all that apply.

#include <iostream>

using namespace std;

_____ ① _____

void foo (_____ ② _____)

{

_____ ③ _____

}

```
—— (4) ——  
int main()  
{  
—— (5) ——  
  for (int i=0; i<10; ++i)      (8) → Variable i have  
  {                             scope : _____  
—— (6) ——  
    }  
—— (7) ——  
}  
}
```

[1] Q8. What is the best prefix for global variables.

[2] Q9. Write a function that return 3 variables.

[1+1+] Q10. Distinguish between initialization and assignment. Give example of both for integer, and for array [Total 4 examples]

[1+] Q11. Give an example that demonstrates variable hiding. If a global variable is hidden, how will you access it in C++?
Is this possible in C?

[3] Q12. Assuming stack can hold 1000×1000 integers, which of the following will result in stack overflow. Why and why not?

(a)

```
int arr[1001][1001];  
int main() { return 0; }
```

(b)

```
int main()  
{ int arr[1001][1001]; }
```

(c)

```
int main()  
{ int *arr = new int[1001*1001]; }
```

[3+1] Q13. (a) Differentiate between "delete" and "delete[]". Give an example that will result in memory leak if proper version of delete is not used.

(b) Pick the odd one out:

(i) new int[5]; (ii) new int(5); (iii) new int {5};

[3] Q14. A double pointer is pointing to some location. Return an int pointer that will point to same location.

```
int* foo(double *d)
{
    // write some lines of code here.
}
```

[5] Q15. By taking inspiration from previous question, print the BIT representation of 5.0 (a double value) on screen. make sure to take care of little endian format.

[Hint: Write a function that prints binary representation of a 1 byte char]

[1+2] Q16. What are default arguments? What are the rules of creating them?

[2] Q17. Why "using namespace std;"?

[5] Q18. Find total number of integer copies that are made in the following program after comment 1 and before comment 2.

```
int foo(int n)
{
    if(n==1) return 1;
    int x = foo(n-1);
    return x;
}

int main()
{
    // comment 1 ←
    int x = foo(2);
    // comment 2 ←
    return 0;
}
```

[3] Q19.

```
Void foo(int *p)
{
    int q=10;
    p = &q;
}
```

```
int main()
{
    int x = 30;
    int *p = &x;
    foo(p);    cout << x << endl;
}
```

 Find output ↑

[3] Q20.

```
int arr[5] = {1, 2, 3, 4, 5};
int *ptr = (int *)(&arr+1);
cout << (*(arr+1)) << " " << (*(ptr-1)) << endl;

```

 ↑ Find output.

[2+2] Q21.
↑ Yes/ no
↑ Explanation

```
Void foo(int arr[])
{
    arr = arr + 1;
    cout << arr[0] << endl;
}
```

```
int main()
{
    int arr[2] = {5, 6};
    foo(arr);
    cout << arr[0] << endl;
}
```

Will the output of above program a garbage value? If yes, why? If no, what is the output?

[3] Q22. Select the correct version of delete that will deallocate memory without memory leak. In case of multiple correct, choose the most appropriate version.

	delete/delete[]	memory (in bytes)
(a) new int [5];	_____	_____
(b) new int (5);	_____	_____
(c) new int {5};	_____	_____

[2+1+2] Q23. (a) What does the following code do?
(b) What is the max value possible for count?
(c) If count = x, how many times operator & is called?

```
int foo(unsigned int n)
{
    int count = 0;
    while (n)
    {
        n = n & (n-1);
        ++count;
    }
}
```

```
    return count;  
}
```

[10] Q24. A delta function is defined as:

```
int delta(char* c, int size)  
{  
    _ _ _ _  
}
```

c is the pointer to first character of string and size is the size of string. for eg

```
char c[5] = {'a', 'b', 'c', 'd', '\0'};  
int x = delta(c, 4);
```

Complete the definition of delta function such that

- ① If s1 can be rearranged to form string s2 (or vice versa), then delta(s1) is same as delta(s2);
- ② Assume that each character is used atmost once in the string and all characters are lowercase english letters.
- ③ Hint: Use Bit manipulation.

[5] Q25.

```
int x = 0x01020304;  
_ _ _ _ 0 _ _ _
```

```
cout << x << endl;
```

 → Prints decimal equivalent of
0x04030201

Use^a character pointer to alter the contents of int present in main memory. Don't allocate any memory in heap.

[1+1+1] Q26. what is void*? Can any pointer be converted to void*? Can void* be converted to any pointer?

[2+1+2] Q27. Complete the following functions:

(a)

```
int updateBit(int num, int i, bool toOne) { }
```

→ If toOne is true, ith bit should be set to 1, else 0.

(b) Use &, <<, ~ operator to clear ith bit of num.

```
int clear(int num, int i) { }
```

(c) what is being done by following functions ?

```
int foo(int num, int i)
{
    return num & ((1 << i) - 1);
}
```

```
int goo(int num, int i)
{
    return num & (-1 << (i + 1));
}
```

marks distribution

Q.	Marks
1	1
2	3
3	6
4	2
5	4
6	6
7	4
8	1
9	2
10	3
11	2
12	3
13	4

Page 1 → 26

Page 2 → 15

Q.	Marks
14	3
15	5
16	3
17	2
18	5
19	3
20	3
21	4
22	3
23	5
24	10
25	5
26	3
27	5

Page 3 → 18

Page 4 → 18

Page 5 → 23

MM: 100