

[Report an Error](#)**Question 1**

Marks for correct response: 1.00 | Negative marking: 0.00

How structures and classes in C++ differ?

- In Structures, members are public by default whereas, in Classes, they are private by default
- In Structures, members are private by default whereas, in Classes, they are public by default
- Structures by default hide every member whereas classes do not
- Structures cannot have private members whereas classes can have

0	Answered	1
29	Not Visited	0
0	Answered & Marked for Review considered for evaluation	

**Choose a Question**

1	2	3
5	6	7
9	10	11
13	14	15
17	18	19
21	22	23
25	26	27
29	30	



**Question 2**

Marks for correct response: 1.00 | Negative marking: 0.00

Which of the following are C preprocessors?

- #ifdef
- #define
- #endif
- all of the mentioned

0	Answered	2
28	Not Visited	0
0	Answered & Marked considered for evaluation	0

Choose a

1	2
5	6
9	10
13	14
17	18
21	22
25	26
29	30



## Question 3

Marks for correct response: 1.00 | Negative marking: 0.00

Which of the following explains Polymorphism?

- int func(int, int);  
 float func1(float, float);
- int func(int);  
 int func(int);
- int func(float);  
 float func(int, int, char);
- int func();  
 int new\_func();

## Question 4

Marks for correct response: 1.00 | Negative marking: 0.00

The C-preprocessors are specified with \_\_\_\_\_ symbol.

- #
- \$
- ..
- &

>

## Question 5

Marks for correct response: 1.00 | Negative marking: 0.00

Which of the following feature of OOPs is not used in the following C++ code?

```
class A
{
int i;
public:
void print(){cout<<"hello"<<i;}
}
```

```
class B: public A
{
int j;
public:
void assign(int a){j = a;}
}
```

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism



## Question 6

Marks for correct response: 1.00 | Negative marking: 0.00

Which of the following is not possible statically in C language?

- Jagged Array
- Rectangular Array
- Cuboidal Array
- Multidimensional Array

>

## Question 7

Marks for correct response: 1.00 | Negative marking: 0.00

Which of the Following Best Describes a Lambda Expression in C++?

- A lightweight function object that can be defined inline
- A pointer to a function
- A preprocessor macro
- An anonymous function



## Question 8

Marks for correct response: 1.00 | Negative marking: 0.00

What is meant by 'a' in the following C operation?

```
fp = fopen("Random.txt", "a");
```

- Attach
- Append
- Apprehend
- Add

>

## Question 9

Marks for correct response: 1.00 | Negative marking: 0.00

What Does the Volatile Keyword do in C++?

- Specifies that a variable cannot be modified
- Specifies that a variable can be modified by external sources
- Specifies that a variable is thread-local
- Specifies that a variable cannot be optimized by the compiler

>

## Question 10

Marks for correct response: 1.00 | Negative marking: 0.00

What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
int y = 10000;
int y = 34;
printf("Hello World! %d\n", y);
return 0;
}
```

- Compile time error
- Hello World! 34
- Hello World! 1000
- Hello World! followed by a junk value

>

## Question 11

Marks for correct response: 1.00 | Negative marking: 0.00

What is the Purpose of the `std::move` Function in C++?

- Moves the contents of one container to another
- Moves the ownership of a resource from one object to another
- Moves the iterator to the next element in a container
- Moves the cursor position in a file

>

## Question 12

Marks for correct response: 1.00 | Negative marking: 0.00

What will happen if the following C code is executed?

```
#include <stdio.h>
int main()
{
int main = 3;
printf("%d", main);
return 0;
}
```

- It will cause a compile-time error
- It will cause a run-time error
- It will run without any error and prints 3
- It will experience infinite looping



## Question 13

Marks for correct response: 1.00 | Negative marking: 0.00

What is a Lambda Capture in C++?

- A mechanism to capture runtime errors
- A way to capture variables from the enclosing scope in a lambda expression
- A method of capturing screenshots in a GUI application
- A technique to capture user input

>

## Question 14

Marks for correct response: 1.00 | Negative marking: 0.00

What will be the output of the following C code?

```
#include <stdio.h>
int main()
{
signed char chr;
chr = 128;
printf("%d\n", chr);
return 0;
}
```

- 128
- 128
- Depends on the compiler
- None of the mentioned

>

## Question 15

Marks for correct response: 1.00 | Negative marking: 0.00

What is the Purpose of the std::forward Function in C++?

- Forwards arguments to another function without moving them
- Moves arguments to another function
- Copies arguments to another function
- Check if arguments are valid

>

## Question 16

Marks for correct response: 1.00 | Negative marking: 0.00

What will be the output of the following C code on a 64 bit machine?

```
#include <stdio.h>
union Sti
{
int nu;
char m;
};
int main()
{
union Sti s;
printf("%d", sizeof(s));
return 0;
}
```

- 8
- 5
- 9
- 4

>

Question 17

Marks for correct response: 1.00 | Negative marking: 0.00

What is the Purpose of the std::async Function in C++?

- Executes a function asynchronously and returns a future object representing the result
- Executes a function synchronously and returns the result immediately
- Executes a function asynchronously without returning a result
- Executes a function synchronously and waits for user input



## Question 18

Marks for correct response: 1.00 | Negative marking: 0.00

What will be the output of the following C function?

```
#include <stdio.h>
enum birds {SPARROW, PEACOCK, PARROT};
enum animals {TIGER = 8, LION, RABBIT, ZEBRA};
int main()
{
    enum birds m = TIGER;
    int k;
    k = m;
    printf("%d\n", k);
    return 0;
}
```

- 0
- Compilation Error
- 1
- 8

>

## Question 19

Marks for correct response: 1.00 | Negative marking: 0.00

What is the Purpose of the Mutable Keyword in a Lambda Expression in C++?

- Specifies that the lambda expression is immutable
- Allows the lambda expression to modify captured variables by value
- Allows the lambda expression to modify captured variables by reference
- Specifies that the lambda expression cannot modify any variables

>

## Question 20

Marks for correct response: 1.00 | Negative marking: 0.00

What will be the output of the following C code?

```
#include <stdio.h>
int const print()
{
printf("Sanfoundry.com");
return 0;
}
void main()
{
print();
}
```

- Error because function name cannot be preceded by const
- Sanfoundry.com
- Sanfoundry.com is printed infinite times
- Blank screen, no output



## Question 21

Marks for correct response: 1.00 | Negative marking: 0.00

What is the Purpose of the std::tie Function in C++?

- Ties two variables together
- Ties function arguments together
- Ties tuple elements to variables
- Ties references to pointers

## Question 22

Marks for correct response: 1.00 | Negative marking: 0.00

Will the following C code compile without any error?

```
#include <stdio.h>
int main()
{
for (int k = 0; k < 10; k++);
return 0;
}
```

- Yes
- No
- Depends on the C standard implemented by compilers
- Error

>

## Question 23

Marks for correct response: 1.00 | Negative marking: 0.00

What is a Move Constructor in C++?

- A constructor that moves an object from one memory location to another
- A constructor that initializes an object using another object's memory
- A constructor that is automatically generated by the compiler
- A constructor that accepts rvalue references as arguments



## Question 24

Marks for correct response: 1.00 | Negative marking: 0.00

What will be the final value of x in the following C code?

```
#include <stdio.h>
void main()
{
int x = 5 * 9 / 3 + 9;
}
```

- 3
- 20
- 24
- None



## Question 25

Marks for correct response: 1.00 | Negative marking: 0.00

What is the Purpose of the std::deque Container in C++?

- A double-ended queue that allows insertion and deletion from both ends
- A stack that follows the Last-In-First-Out (LIFO) principle
- A queue that follows the First-In-First-Out (FIFO) principle
- A set that stores unique elements in sorted order



## Question 26

Marks for correct response: 1.00 | Negative marking: 0.00

What will be the output of the following C code? (Initial values: x= 7, y = 8)

```
#include <stdio.h>
void main()
{
float x;
int y;
printf("enter two numbers \n");
scanf("%f %f", &x, &y);
printf("%f, %d", x, y);
}
```

- 7.000000, 7
- Run time error
- 7.000000, junk
- Varies

>

## Question 27

Marks for correct response: 1.00 | Negative marking: 0.00

Find the odd one out.

- std::vector<int>
- std::vector<short>
- std::vector<long>
- std::vector<bool>



C

3

0

1

2

3

4

## Question 28

Marks for correct response: 1.00 | Negative marking: 0.00

What will be the output of the following C code considering the size of a short int is 2, char is 1 and int is 4 bytes?

```
#include <stdio.h>
int main()
{
short int i = 20;
char c = 97;
printf("%d, %d, %d\n", sizeof(i), sizeof(c), sizeof(c + i));
return 0;
}
```

- 2,1,2
- 2,1,1
- 2,1,4
- 2,2,8

>

## Question 29

Marks for correct response: 1.00 | Negative marking: 0.00

Which of the following statements are false?

- bool can have two values and can be used to express logical expressions
- bool cannot be used as the type of the result of the function
- bool can be converted into integers implicitly
- a bool value can be used in arithmetic expressions

>

0

1

0

9

10

17

21

25

29

## Question 30

Marks for correct response: 1.00 | Negative marking: 0.00

What is the difference between the following 2 C codes?

Code - 1:

```
#include <stdio.h> //Program 1
int main()
{
int d, a = 1, b = 2;
d = a++ + ++b;
printf("%d %d %d", d, a, b);
}
```

Code - 2:

```
#include <stdio.h>
int main()
{
int d, a = 1, b = 2;
d = a++ ++b;
printf("%d %d %d", d, a, b);
}
```

- No difference as space doesn't make any difference, values of a, b, d are same in both the case
- Space does make a difference, values of a, b, d are different