



Software Engineering

Home ► My courses ► Previous Years ► 2020-21 ► Spring Semester (2020-21) ►
Software Engineering ► Theory Quiz - 1 ► Theory Quiz: 1

Started on Wednesday, 27 January 2021, 2:15 PM

State Finished

Completed on Wednesday, 27 January 2021, 3:07 PM

Time taken 52 mins 8 secs

Grade 15.00 out of 15.00 (100%)

Question 1

Correct

Mark 1.00 out of
1.00

Flag question

Consider the following program.

```
#include <iostream>
using namespace std;

int main() {
    int i = 0;
    cout << sizeof(&i) << " ";
    cout << sizeof(void*);

    return 0;
}
```

What are the possible outputs of the above program?

Select one or more:

☐

4 8

☒

8 8

☒

4 4





8 4

Your answer is correct.

The correct answer is:

8 8

,

4 4

Question 2

Correct

Mark 2.00 out of 2.00

Flag question

Consider the program below where the declarator and qualifer for variable `i` is missing.

```
#include <iostream>
#include <cmath>
using namespace std;

int& func(int& a) {
    _____ i = a; // Declarator needed

    return i;
}

int main() {
    int a = 16;
    int& b = func(a);
    a = sqrt(sqrt(sqrt(sqrt(a))));

    cout << b + a << endl;

    return 0;
}
```

Given below are different fill-up options for the declarator,

Option # Declarator

- (a) `int`
- (b) `int&`
- (c) `const int`
- (d) `static int`

and the outcome from the program as filled up.

Option # Outcome

- (1) Compilation Error
- (2) 17
- (3) Segmentation fault / Indeterminate value
- (4) 2

Match every fill-up option with the corresponding outcome.

const int ✓

int ✓

int& ✓

static int ✓

Your answer is correct.

The correct answer is: const int

- compilation error, int
- Segmentation fault / Indeterminate value, int&
- 2, static int
- 17

Question 3

Correct

Mark 0.50 out of 0.50

Flag question

Choose the equivalent expression using operator function/s for the expression in the following statement:

`c = a + b ;`

Select one:

☐

`operator+(operator=(c, a), b);`

☐

`=(c, +(a, b));`



operator=(c, add(a, b));



operator=(c, operator+(a, b));



Your answer is correct.


The correct answer is:

operator=(c, operator+(a, b));

Question 4

Correct

Mark 1.00 out of 1.00

 Flag question

What will be the output of the following program?

```
#include <iostream>
using namespace std;

class num {
    int a;
    static int n;
public:
    num() : a(0) {
        ++n;
        a += n % 2 * n;
        cout << "a=" << a << " " << "n=" << n << endl;
    }
};

int num::n = 0;

int main() {
    num x, a[2];

    return 0;
}
```

Select one:



a=1 n=1
a=0 n=2
a=3 n=3





```
a=1 n=1
a=1 n=1
a=1 n=1
```



```
a=1 n=1
a=2 n=2
a=3 n=3
```



```
a=1 n=1
a=0 n=2
a=1 n=3
```

Your answer is correct.


The correct answer is:

```
a=1 n=1
a=0 n=2
a=3 n=3
```

Question 5

Correct

Mark 1.00 out of 1.00

 Flag question

Match the purpose with the mechanism between the lists.

Option # Purpose

- (a) Efficient transfer of value from caller to callee
- (b) Reliable transfer of value of local variable from callee to caller
- (c) Necessary to use a function as an lvalue
- (d) Safe transfer of value from caller to callee

and the outcome from the program as filled up.

Option # Mechanism

- (1) Call-by-Value
- (2) Call-by-Reference
- (3) Return-by-Value
- (4) Return-by-Reference

Efficient transfer of value from caller to callee

Call-by-Reference 



Reliable transfer of value of local variable from callee to caller

Return-by-Value ▼



Necessary to use a function as an lvalue

Return-by-Reference ▼



Safe transfer of value from caller to callee

Call-by-Value ▼



Your answer is correct.

The correct answer is: Efficient transfer of value from caller to callee

- Call-by-Reference, Reliable transfer of value of local variable from callee to caller
- Return-by-Value, Necessary to use a function as an lvalue
- Return-by-Reference, Safe transfer of value from caller to callee
- Call-by-Value

Question 6

Correct

Mark 1.00 out of 1.00

Flag question

What will be the output of the following program?

```
#include <iostream>
using namespace std;

#define MM_1(x, y) x * y
#define MM_2(x, y) (x * y)
#define MM_3(x, y) (x) * (y)

inline int MPLY(int x, int y) { return x * y; }

int main() {
    int a = 3, b = 4;

    cout << "MM_1:" << MM_1(MM_1(a + 1, b - 1), a + b) << endl;
    cout << "MM_2:" << MM_2(MM_2(a + 1, b - 1), a + b) << endl;
    cout << "MM_3:" << MM_3(MM_3(a + 1, b - 1), a + b) << endl;
    cout << "MPLY:" << MPLY(MPLY(a + 1, b - 1), a + b) << endl;

    return 0;
}
```

Select one:



MM_1:8
MM_2:22
MM_3:84
MPLY:84



MM_1:8
MM_2:8
MM_3:84
MPLY:84



MM_1:8
MM_2:22
MM_3:22
MPLY:84



MM_1:8
MM_2:84
MM_3:84
MPLY:84

Your answer is correct.

The correct answer is:

MM_1:8
MM_2:22
MM_3:84
MPLY:84

Question 7

Correct

Mark 1.00 out of
1.00

Flag question

Consider the following program.

```
int main() {  
    int a[] = { 1, 2, 3, 4, 5 };  
    int *arr[5] = { a, a + 1, a + 2, a + 3, a + 4 };  
    int *p = a + 4;  
  
    cout << p[*arr[1] - 5];  
  
    cin >> *p;  
    return 0;  
}
```

What will be the output ?

Select one:

☒

2

☐

1

☐

garbage

☐

3

Your answer is correct.


The correct answer is:

2

Question 8

Correct

Mark 0.50 out of
0.50

 Flag question

What is the output of the following program/

```
#include<iostream>
using namespace std;

class xyz {
public:
    xyz() {
        cout << "Hi" << endl;
    }
} p;

int main() {
    cout << "I am in main" << endl;

    return 0;
}
```

Select one:



Compilation Error



Segmentation fault



Hi
I am in main



I am in main

Your answer is correct.


The correct answer is:

Hi
I am in main

Question 9

Correct

Mark 1.00 out of
1.00

 Flag question

What will be the output of the following program?

```
#include <iostream>
using namespace std;

int i = 0;

class A {
public:
    ~A() { i = 10; }
};

int fun() {
    i = 3;
    A obj;

    return i++;
}

int main() {
    cout << fun() << " ";
    cout << i << endl;

    return 0;
}
```

Select one:

☒

3 10

☐

10 11

☐

4 10

☐

3 4


Your answer is correct.

The correct answer is:

3 10

Question 10

Correct

Mark 0.50 out of
0.50 Flag question

Consider class `Test`. What are the permissible signatures of a Copy Constructor?

Select one:


☐`Test(const Test& t), Test(Test& t);`☐`Test(const Test t), Test(Test t);`☐`Test(const Test* t), Test(Test* t);`☐`Test(Test& t), Test(Test* t);`

Your answer is correct.

The correct answer is:

`Test(const Test& t), Test(Test& t);`**Question 11**

Correct

Mark 1.00 out of
1.00 Flag question

What will be the output from the program below?

```
#include <iostream>
using namespace std;

class sample {
    int i;
    double d;
public:
    sample(int i_, double d_) : i(i_), d(d_) { }
    void print(int i, double d) {
        cout << i << " " << d << endl;
        cout << this->i << " " << this->d << endl;
    }
};

int main() {
    sample s(2, 3);

    s.print(5, 6);

    return 0;
}
```

Select one:

☒

5 6
2 3


☐

2 3
5 6

☐

2 3
2 3

☐

5 6
5 6

Your answer is correct.


The correct answer is:

5 6
2 3

Question 12

Correct

Mark 2.00 out of 2.00

 Flag question

For a singleton class `QueenOfEngland` below, we have an application in `main()`. Two lines of the class (line-1 and line-2) are partially blank and need to be filled up properly.

```
#include <iostream>
using namespace std;

class QueenOfEngland {
    static QueenOfEngland* _myQueen;
    static int reign;
    QueenOfEngland() { reign++; }
public:
    const static QueenOfEngland& queen() {
        if (!_myQueen)
            _myQueen = _____; // line-1
        return _____; // line-2
    }
    void Reign() const { cout << reign << " "; }
};

QueenOfEngland* QueenOfEngland::_myQueen = NULL;
int QueenOfEngland::reign = 0;

int main() {
    QueenOfEngland::queen().Reign();
    QueenOfEngland::queen().Reign();
    QueenOfEngland::queen().Reign();

    return 0;
}
```

Given below are different fill-up options,

Options#	line-1	line-2
(a)	<code>new QueenOfEngland()</code>	<code>_myQueen</code>
(b)	<code>new QueenOfEngland()</code>	<code>*_myQueen</code>
(c)	<code>new QueenOfEngland()</code>	<code>*new QueenOfEngland()</code>
(d)	<code>NULL</code>	<code>*new QueenOfEngland()</code>

and the outcome from the program as filled up.

Option # Outcome

(1) 2 3 4

Option # Outcome

- (2) Compilation Error
(3) 1 2 3
(4) 1 1 1

Match every fill-up option with the corresponding outcome.

- (a) ✓
(c) ✓
(d) ✓
(b) ✓

Your answer is correct.

The correct answer is: (a)

- Compilation Error, (c)
- 2 3 4, (d)
- 1 2 3, (b)
- 1 1 1

Question 13

Correct

Mark 1.00 out of
1.00

Flag question

Consider the following program.

```
#include <stdio>
#include <stdlib>
using namespace std;

int main() {
    int *p = (int*)malloc(sizeof(int));
    *p = 0x5E6A3D1B;
    unsigned char *q = (unsigned char*)p;
    printf("%X\n", *(q + 1));

    return 0;
}
```

What will be the output ?

Select one:



3D





6A



1B



5E

Your answer is correct.


The correct answer is:

3D

Question 14

Correct

Mark 1.00 out of
1.00

 Flag question

What will be the output of the program below?

```
#include <iostream>
using namespace std;

class student {
    int mark;
public:
    student(): mark(0) { }
    student(int x): mark(x) { }
    student(const student& s): mark(s.mark) { }
    ~student() { cout << mark << " "; }
};

int main() {
    student s1(200);
    student s2();
    student s3 = 300;
    student s4 = s3;

    return 0;
}
```

Select one:



Syntax error in declaring object



300 300 200

☐

200 300 300

☐

300 300 0 200

Your answer is correct.


The correct answer is:

300 300 200

Question 15

Correct

Mark 0.50 out of
0.50

 Flag question

What is INCORRECT about static member functions?

Select one:

☐

can be of private or public access type

☐

is shared by all objects of a class

☒

can access both static and non-static members

☐

does not have this pointer

Your answer is correct.

The correct answer is:

can access both static and non-static members

Finish review

QUIZ NAVIGATION

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#)

Show one page at a time

Finish review

You are logged in as Nisarg Upadhyaya (Log out)
CS20006_S2021