X

Week 10 : Assignment 10

Your last recorded submission was on 2025-04-01, 19:51 Due date: 2025-04-02, 23:59 IST. IST

1) ***1 point***

Consider the code segment (in C++11) given below.

```
#include<iostream>
int main( ){
    char n = 'A';
    char& rn = n;
    ___________ t = rn;    //LINE-1

    ++t;
    std::cout << n << " " << rn << " " << t << std::endl;
    return 0;
}
```

Identify the appropriate option/s to fill in the blank at LINE-1 such that output becomes B B B

a) auto

b) auto&

c) decltype(rn)

d) decltype(n)

- [ ] a.
- [x] b.
- [x] c.
- [ ] d.

2) ***1 point***

Consider the code segment (in C++14) given below.

```
#include<iostream>
struct Oper1{
    int i;
    Oper1(int _i) : i(_i){}
    int& operator()() { std::cout << "1 "; return i ; }
};

struct Oper2{
    int i;
    Oper2(int _i) : i(_i){}
    int operator()() { std::cout << "2 "; return i ; }
};


template < typename U >
------------------ {    //LINE-1
    return op() ;
}

int main(){
    Oper1 o1{10};
    Oper2 o2{10};
    foobar(o1) = 20;
    foobar(o2) ;
    return 0;
}
```

Identify the appropriate option/s to fill in the blank at `LINE-1` such that output becomes `1 2`.

a) `auto foobar( U& op ) -> decltype(op())`

b) `auto foobar( U& op )`

c) `auto& foobar( U& op )`

d) `decltype(auto) foobar( U& op )`

- [x] a.
- [ ] b.
- [ ] c.
- [x] d.

3) ***1 point***

Consider the following class (int C++11).

```
class CustList{
    public:
        CustList(std::initializer_list<double> dlist) { cout << "ctor-1" << " "; }
        CustList(std::initializer_list<int> dlist) { cout << "ctor-2" << " "; }
        CustList(double d1, double d2, double d3) { cout << "ctor-3" << " "; }
};
```

Indemnify the appropriate option that present all correct output/error for the following instantiation of `CustList` class:

1. `CustList c{3.1, 4.5, 6.5};`
2. `CustList c(3.1, 4.5, 6.5);`
3. `CustList c{3.1f, 4.5f, 6.5};`
4. `CustList c{3, 4.5, 6};`

a) (a) `ctor-1`
(b) `ctor-3`
(c) `ctor-1`
(d) `ctor-3`

b) (a) `ctor-1`
(b) `ctor-3`
(c) `ctor-1`
(d) `compiler error: call of overloaded 'CustList()' is ambiguous for CustList c{3, 4.5, 6};`

c) (a) `ctor-1`
(b) `ctor-1`
(c) `ctor-1`
(d) `compiler error: call of overloaded 'CustList()' is ambiguous for CustList c{3, 4.5, 6};`

d) (a) `ctor-3`
(b) `ctor-1`
(c) `compiler error: call of overloaded 'CustList()' is ambiguous for CustList c{3.1f, 4.5f, 6.5};`
(d) `compiler error: call of overloaded 'CustList()' is ambiguous for CustList c{3, 4.5, 6};`

- ○ a.
- ◉ b.
- ○ c.
- ○ d.

4) ***1 point***

Books ()

Transcripts ()

Problem Solving Session - Jan 2025 ()

Consider the following code segment.

```
#include <iostream>

class ComplexNum{
    public:
        constexpr ComplexNum(int _r = 0, int _i = 0) : r(_r), i(_i){ }
    private:
        int r, i;
};
int randGen(){
    return 10;
}
constexpr int numGen(int i, int j){
    return i + j;
}
int main(){
    constexpr ComplexNum c1(10, 20);    //LINE-1
    constexpr int i = 10, j = 20;
    constexpr ComplexNum c2(i, j);      //LINE-2
    constexpr ComplexNum c3(randGen(), randGen());          //LINE-3
    constexpr ComplexNum c4(numGen(i, j), numGen(i, j));    //LINE-4
    return 0;
}
```

Which of the following line/s generate/s compiler error?

a) LINE-1

b) LINE-2

c) LINE-3

d) LINE-4

- [ ] a.
- [ ] b.
- [x] c.
- [ ] d.

5) Consider the following code segment. **1 point**

```
#include <iostream>

void update(char* str){ /*some code*/ }

template<typename F, typename P>
void caller(F func, P s){
    func(s);
}

int main(){
    char s[2] = "0";
    char *p = &s[1];
    caller(update, p);          //LINE-1
    caller(update, 0);          //LINE-2
    caller(update, NULL);       //LINE-3
    caller(update, nullptr);    //LINE-4
    return 0;
}
```

Which of the following lines generate/s compiler error?

a) LINE-1

b) LINE-2

c) LINE-3

d) LINE-4

- [ ] a.
- [x] b.
- [x] c.
- [ ] d.

6) **1 point**

Consider the following code segment (in C++11).

```
#include<iostream>
#include<iomanip>

---------------------------------------- {    //LINE-1
   return 1024 * mem;
}

---------------------------------------- {    //LINE-2
   return mem;
}
int main() {
   long double size = 10.0_KB + 2.0_B;
   std::cout << "size (in bytes): " << size;
   return 0;
}
```

Identify the appropriate option to fill in the blanks at LINE-1 and LINE-2 such that the output becomes size (in bytes): 10242.

a)
```
LINE-1:  long double operator"" KB(long double mem)
LINE-2:  long double operator"" B(long double mem)
```

b)
```
LINE-1:  long double operator"" _KB(long double mem)
LINE-2:  long double operator"" _B(long double mem)
```

c)
```
LINE-1:  long int operator"" _KB(long int mem)
LINE-2:  long int operator"" _B(long int mem)
```

d)
```
LINE-1:  unsigned long long operator _KB(unsigned long long mem)
LINE-2:  unsigned long long operator _B(unsigned long long mem)
```

- ○ a.
- ◉ b.
- ○ c.
- ○ d.

7) ***1 point***

Consider the code segment (in C++11) below.

```
#include<iostream>
#include<vector>
#include<cmath>

void process(int& v) {
    if(v < 0)
        throw v;
    ++v;
}

void func(std::vector<int>& iVec) noexcept(noexcept(process(iVec[0]))){
    for(int& v : iVec)
        process(v);
}

int main() {
    std::vector<int> iVec{1, 2, -1, 2};
    try{
        func(iVec);
    }catch(int i){
    }
    for(int v : iVec)
        std::cout << v << " ";
    return 0;
}
```

Identify the correct option about the program above.

a) It generates output as `2 3 0 3`

b) It generates output as `2 3 -1 2`

c) It generates output as `2 3 -1 3`

d) The program gets terminated since a function that is declared `noexcept` throws an exception

- ○ a.
- ◉ b.
- ○ c.
- ○ d.

8) Consider the following code segment (in C++11). **1 point**

```
#include<iostream>
int i = 10;

void test(int&& rv){ }

int getVal(){
    return i;
}

int& getRef(){
    return i;
}

int main() {
    test(i);            //LINE-1
    test(i + 10);       //LINE-2
    test(getVal());     //LINE-3
    test(getRef());     //LINE-4
    return 0;
}
```

Identify the line/s generate/s compiler error.

a) LINE-1

b) LINE-2

c) LINE-3

d) LINE-4

- [x] a.
- [ ] b.
- [ ] c.
- [x] d.

9) **1 point**

Consider the following code segment (in C++ 11).

```
#include<iostream>

class Resource {
    public:
        Resource() { std::cout << "#1" << " "; }
        Resource(const Resource&) { std::cout << "#2" << " "; }
        Resource(Resource&&) noexcept { std::cout << "#3" << " "; }
        Resource& operator=(const Resource&) { std::cout << "#4" << " ";
                                return *this; }
        Resource& operator=(Resource&&) noexcept { std::cout << "#5" << " ";
                                return *this; }
};

Resource createResource(){
    Resource r;
    return r;
}

int main() {
    Resource r1;
    r1 = createResource();
    Resource r2 = r1;
    Resource r3 = std::move(r2);
    return 0;
}
```

What will be the output?

a) #1 #5 #4 #3

b) #1 #1 #5 #4 #3

c) #1 #3 #2 #3

d) #1 #1 #5 #2 #3

- ○ a.
- ○ b.
- ○ c.
- ◉ d.

You may submit any number of times before the due date. The final submission will be considered for grading.

Submit Answers