# Thank you for taking the Week 10 : Assignment 10.

## Week 10 : Assignment 10

**Your last recorded submission was on 2023-03-31, 11:58 Due date: 2023-04-05, 23:59 IST. IST**

1) ***2 points***

Assessment submitted.
X



Consider the program (in C++11) given below.

```
#include <iostream>

__________________________ {            // LINE-1
    double dct = 0.05;
    double getDct(double pri){
        return pri * dct;
    }
}

__________________________ {            // LINE-2
    double dct = 0.07;
    template<typename T>
    T getDct(T pri){
        return pri * dct;
    }
}

int main(){
   std::cout << ver1_0::getDct(105.0) << " ";
   std::cout << ver2_0::getDct(105) << " " << getDct(105.0);
   return 0;
}
```

Choose the appropriate option to fill in the blanks at `LINE-1` and `LINE-2` so that the output becomes
`5.25 7 7.35`

a)
```
LINE-1:  namespace ver1_0
LINE-2:  namespace ver2_0
```

b)
```
LINE-1:  namespace ver1_0
LINE-2:  inline namespace ver2_0
```

c)
```
LINE-1:  inline namespace ver1_0
LINE-2:  namespace ver2_0
```

d)
```
LINE-1:  inline namespace ver1_0
LINE-2:  inline namespace ver2_0
```

- (●) a)
- ( ) b)
- ( ) c)
- ( ) d)

2) ***2 points***

Assessment submitted.
X

(/noc23_cs50/progassignment?name=209)
W10_Programming_Qs-2 (/noc23_cs50/progassignment?name=210)
W10_Programming_Qs-3 (/noc23_cs50/progassignment?name=211)
Week 10 Feedback Form (unit?unit=112&lesson=120)
**Download Videos ()**
**Books ()**
**Transcripts ()**
**Problem Solving Session ()**

Consider the program (in C++11) given below.

```
#include <iostream>

int main( ){
    int n1 = 10;
    const int n2 = 10;

    int& i1 = n1;
    const int& i2 = n2;

    auto x1 = i1;
    auto x2 = i2;                                   //LINE-1

    std::cout << ++x1 << " " << ++x2 << " ";        //LINE-2
    std::cout << i1 << " " << i2;

    return 0;
}
```

What will be the output/error?

a) Compiler error at LINE-1: auto connot deduce to cv-qualifier

b) Compiler error at LINE-2: read-only x2 cannot be modified

c) 11 11 11 11

d) 11 11 10 10

- ○ a)
- ○ b)
- ○ c)
- ◉ d)

3) ***2 points***

Assessment submitted.
X

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

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

void change(std::vector<int>& iv){
    auto j = 10;
    for(_____________ : iv)   //LINE-1
        i += j;
}
int main( ){
    std::vector<int> iVec { 10, 20, 30, 40 };
    change(iVec);
    for(auto i : iVec)
        std::cout << i << ", ";
    return 0;
}
```

Choose the appropriate option/s to fill in the blank at `LINE-1` such that the output is `20, 30, 40, 50,`

a) `auto i`

b) `decltype(j) i`

c) `decltype((j)) i`

d) `decltype(iv[j]) i`

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

4) ***2 points***

Assessment submitted.
X

Consider the following code segment (in C++14).

```
#include<iostream>

struct LFunc{
    int i {10};
    int operator()() { return i ; }
};

struct RFunc{
    int i {10};
    int& operator()() { return i ; }
};


template < typename T >
------------------------- {    //LINE-1
    return rf() ;
}

int main(){
    LFunc f1;
    RFunc f2;
    std::cout << caller(f1) << " ";
    std::cout << (caller(f2) = 20);
    return 0;
}
```

Choose the appropriate option/s to fill in the blank at `LINE-1` such that the output is `10 20`.

a) `auto caller( T& rf )`

b) `auto caller( T& rf ) -> decltype(rf())`

c) `int& caller( T& rf )`

d) `decltype(auto) caller( T& rf )`

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

5) ***2 points***

Assessment submitted.
X

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

```
#include <iostream>

class point{
    public:
        constexpr point(int x = 0, int y = 0) : x_(x), y_(y){ }
    private:
        int x_, y_;
};
int genN(){
    return 10;
}
constexpr int genN(int i, int j){
    return i + j;
}
int main(){
    constexpr point p1 {100, 200};              //LINE-1
    constexpr int i = 10;
    int j = 20;
    constexpr point p2 {i, j};                  //LINE-2
    constexpr point p3(genN(), genN());         //LINE-3
    constexpr point p4(genN(i, j), genN(i, j)); //LINE-4
    return 0;
}
```

Identify the function call/s that will compile without generating any error.

a) LINE-1

b) LINE-2

c) LINE-3

d) LINE-4

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

6) ***2 points***

Assessment submitted.
X

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

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

long double operator"" _KM(long double x) {
    return x * 1000;
}
long double operator"" _M(long double x) {
    return x;
}
class distance{
    public:
        distance(int d1, int d2) : d1_(d1), d2_(d2){}
        int getDistance(){ return d1_ + d2_; }
    private:
        int d1_, d2_;
};
int main() {
    distance d(________________________);    //LINE-1
    std::cout << d.getDistance() << "M";
    return 0;
}
```

Choose the appropriate option to fill in the blank at `LINE-1` such that the output is `5011M`.

a) `5.0KM, 11.0M`

b) `5.0_KM, 11.0_M`

c) `(KM)5.0, (M)11.0`

d) `5_KM, 11_M`

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

7) ***2 points***

Assessment submitted.
X

Consider the following program (in C++11).

```
#include<iostream>

class base{
    public:
        base(const int& x) { std::cout << "#1 " ; }
        base(const base& ob) { std::cout << "#2 " ; }
        base(base&& ob) noexcept { std::cout << "#3 " ; }
};

class derived : public base {
    public:
        derived(const int& x, const int& y) : base(x) { std::cout << "#4 " ; }
        derived(const derived& ob) : base(ob) { std::cout << "#5 " ; }
        derived(derived&& ob) noexcept : base(ob) { std::cout << "#6 " ; }
};

int main(){
    derived o1(100, 200);
    derived o2(o1);
    derived o3(std::move(o1));
    return 0;
}
```

What will be the output?

a) #1 #4 #2 #5 #3 #6

b) #1 #4 #2 #5 #2 #6

c) #1 #4 #2 #5 #3 #5

d) #1 #4 #2 #6 #2 #6

- (●) a)
- ( ) b)
- ( ) c)
- ( ) d)

8) ***2 points***

Assessment submitted.
X

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

```
#include <iostream>
#include <list>
#include <initializer_list>

class items{
    public:
        items(int n) { std::cout << "#1 " ; }
        items(std::initializer_list<int> vals) { std::cout << "#2 " ; }
        items(int n, std::initializer_list<int> vals) { std::cout << "#3 " ; }
};
int main(){
    items i1(10);               //LINE-1
    items i2{10, 20, 30};       //LINE-2
    items i3({10});             //LINE-3
    items i4{10};               //LINE-4
    items i5 = {10};            //LINE-5
    items i6(10, {10, 20, 30}); //LINE-6
    return 0;
}
```

What will be the output?

a) #1 #2 #1 #1 #1 #3

b) #1 #2 #1 #1 #1 #2

c) #1 #2 #1 #2 #2 #3

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

- ○ a)
- ○ b)
- ○ c)
- ◉ d)

9) ***2 points***

Assessment submitted.
X

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

```
#include <iostream>

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

template<typename FUNC, typename PARA>
void wrapper(FUNC f, PARA p){
    f(p);
}

int main(){
    char s[4] = "C++";
    wrapper(print, s);          //LINE-1
    wrapper(print, 0);          //LINE-2
    wrapper(print, s[4]);       //LINE-3
    wrapper(print, nullptr);    //LINE-4
    return 0;
}
```

Choose the call/s to `wrapper` function that will result in compiler error/s.

a) `LINE-1`

b) `LINE-2`

c) `LINE-3`

d) `LINE-4`

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

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

**Submit Answers**