

# DUMPS ARENA

**C++ Certified Professional Programmer**

C++ Institute CPP

Version Demo

**Total Demo Questions: 15**

**Total Premium Questions: 228**

**Buy Premium PDF**

<https://dumpsarena.com>

[sales@dumpsarena.com](mailto:sales@dumpsarena.com)

**dumpsarena.com**

Topic Break Down

Topic	No. of Questions
Topic 1, Volume A	110
Topic 2, Volume B	118
Total	228

**QUESTION NO: 1**

Which pieces of code inserted independently into places marked 1 and 2 will cause the program to compile and display: 0 1 2 3 4 5 6 7 8 9? Choose all that apply.

```
#include #include using namespace std; class A { int a; public: A(int a){ this->a=a;}
```

```
//insert code here 1
```

```
};
```

```
//insert code here 2
```

```
template void print(T start, T end) { while (start != end) {
```

```
std::cout << *start << " "; start++;
```

```
}
```

```
}
```

```
int main() {
```

```
A t1[] ={ 1, 7, 8, 4, 5 };list l1(t1, t1 + 5); A t2[] ={ 3, 2, 6, 9, 0 };list l2(t2, t2 + 5); l1.sort();l2.sort();l1.merge(l2); print(l1.begin(), l1.end());
```

```
print(l2.begin(), l2.end()); cout<
```

**A.** place 1: operator int() { return a; }

**B.** place 1: operator int() { return a; } bool operator < (const A & b) { return this->a< b.a;}

**C.** place 1: bool operator < (const A & b) { return this->a< b.a;}

**D.** place 1: bool operator < (const A & b) { return this->a< b.a;} friend ostream & operator <<(ostream & c, const A & a); place 2: ostream & operator <<(ostream & c, const A & a) { c<

**E.** place 1: bool operator < (const A & b) { return this->a< b.a;} place 2: ostream & operator <<(ostream & c, const A & a) { c<

**ANSWER: A B D****QUESTION NO: 2**

Which changes introduced independently will allow the code to compile and display 0 0 1 1 8 8 9 9 (choose all that apply)?

```
#include
```

```
#include #include using namespace std;
```

```
class A { int a; public:
```

```
A(int a):a(a){  
int getA() const { return a;}  
/* Insert Code Here 1 */  
};  
/* Insert Code Here 2*/  
int main(){  
A t[]={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 }; sets(t, t+10);/* Replace Code Here 3 */  
multiset s1(s.begin(),s.end());/* Replace Code Here 4 */ s1.insert(s.begin(),s.end());  
s1.erase(s1.lower_bound(2),s1.upper_bound(7)); multiset::iterator i=s1.begin();/* Replace Code Here 5 */ for( ;i!= s1.end();  
i++)  
{  
cout<getA()<<" ";  
}  
cout<<<>  
return 0; }
```

**A.** operator int() const { return a;} inserted at Place 1

**B.** bool operator < (const A & b) const { return a

**C.** bool operator < (const A & b) const { return b.a

**D.** struct R { bool operator()(const A & a, const A & b) { return a.getA()

replacing line marked 3 with sets(t, t+10);

replacing line marked 4 with multiset s1(s.begin(),s.end()); replacing line marked 5 with multiset::iterator i=s1.begin();

**ANSWER: A B D**

### QUESTION NO: 3

What will happen when you attempt to compile and run the following code?

```
#include
```

```
#include
```

```
#include
```

```
#include #include using namespace std; int main() {
```

```
int t[] = { 3, 4, 2, 1, 0, 3, 4, 1, 2, 0 }; vector v(t, t + 10); multimap m;
```

```
for (vector::iterator i = v.begin(); i != v.end(); i++) { stringstream s;s << *i << *i;
m.insert(pair(*i, s.str()));
}
pair<>::iterator, multimap::iterator> range; range = m.equal_range(2);
for (multimap::iterator i = range.first; i != range.second; i++) { cout << i->first << " ";
} return 0;
}
```

The output will be:

- A. 2 2
- B. 1 2
- C. 1 3
- D. 2
- E. 0 2

**ANSWER: A**

#### QUESTION NO: 4

What will happen when you attempt to compile and run the following code?

```
#include
#include
using namespace std;
int main ()
{
float f = 10.126; cout.unsetf(ios::floatfield);
cout<<><><><>
```

What will be a mantissa part of the numbers displayed:

- A. 1.0126 1.013
- B. 1.012600 10.013
- C. 10.01260 10.013
- D. 1.012600 1.013

E. 1.0126 1.01

**ANSWER: D**

**QUESTION NO: 5**

What happens when you attempt to compile and run the following code?

```
#include
```

```
#include
```

```
#include #include using namespace std; templatestruct Out { ostream & out;
```

```
Out(ostream & o): out(o){}
```

```
void operator() (const T & val ) { out<
```

```
int t[]={8, 10, 5, 1, 4, 6, 2, 7, 9, 3}; deque d1(t, t+10); set s1(t,t+10);
```

```
cout<
```

```
<<>
```

```
return 0; }
```

Choose all possible outputs (all that apply):

**A.** 1 0

**B.** 1 1

**C.** true true

**D.** false false

**E.** compilation error

**ANSWER: A B**

**QUESTION NO: 6**

Which stack initialization (line numbers) are correct? Choose all that apply.

```
#include
```

```
#include
```

```
#include
```

```
#include #include
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
deque mydeck; list mylist; vector myvector; stack first;// Line I stack second(mydeck);// Line II stack third(second);// Line III  
stack > fourth(mylist);// Line IV stack > fifth(myvector);// Line V return 0; }
```

- A. line I
- B. line II
- C. line III
- D. line IV
- E. line V

**ANSWER: A B C D E**

#### QUESTION NO: 7

What happens when you attempt to compile and run the following code?

```
#include
```

```
#include
```

```
#include #include
```

```
using namespace std;
```

```
void myfunction(int i) {
```

```
cout << " " << i;
```

```
}
```

```
int main() {
```

```
int t[] = { 10, 5, 9, 6, 2, 4, 7, 8, 3, 1 }; deque d1(t, t+10); vector v1(d1.rbegin(), d1.rend()); sort(d1.begin(), d1.end());
```

```
swap_ranges(v1.begin(), v1.end(), d1.begin()); for_each(v1.begin(), v1.end(), myfunction); for_each(d1.begin(), d1.end(),  
myfunction); return 0; }
```

Program outputs:

- A. 10 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9 10
- B. compilation error
- C. 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10
- D. 1 2 3 4 5 6 7 8 9 10 1 3 8 7 4 2 6 9 5 10

E. 1 3 8 7 4 2 6 9 5 10 1 2 3 4 5 6 7 8 9 10

**ANSWER: D**

### QUESTION NO: 8

What happens when you attempt to compile and run the following code?

```
#include  
#include  
#include #include using namespace std; template struct Out { ostream & out; Out(ostream & o): out(o){}  
void operator() (const T & val ) { out<  
int start;  
Sequence(int start):start(start){}  
int operator()() { return start++; } };  
int main() { vector v1(10);  
generate_n(v1.begin(), 10, Sequence(1)); random_shuffle(v1.rbegin(), v1.rend()); sort(v1.begin(), v1.end(), great());  
for_each(v1.begin(), v1.end(), Out(cout));cout<
```

Program outputs:

- A. 8 10 5 1 4 6 2 7 9 3
- B. 1 2 3 4 5 6 7 8 9 10
- C. compilation error
- D. 10 9 8 7 6 5 4 3 2 1

**ANSWER: C**

### QUESTION NO: 9

What happens when you attempt to compile and run the following code?

```
#include  
#include  
#include #include using namespace std;  
bool identical(int a, int b) {
```



```
return b == 2*a?true:false;

}

int main() {

int t[] = {1,2,3,2,3,5,1,2,7,3,2,1,10, 4,4,5}; int u[] = {2,4,6,4,6,10,2,4,14,6,4,2,20,8,8,5}; vector v1(t, t + 15);

deque d1(u, u + 15);

pair<>::iterator, vector::iterator > result; result = mismatch(d1.begin(), d1.end(), v1.begin(), identical); //Line I if (result.first ==
d1.end() && result.second == v1.end()) { //Line II cout<<"Identical\n";

} else {

cout<<"Not identical\n";

} return 0; }
```

Program outputs:

- A. Identical
- B. Not identical
- C. compilation error at line marked I
- D. compilation error at line marked II

**ANSWER: B**

#### QUESTION NO: 10

What happens when you attempt to compile and run the following code?

```
#include

#include #include using namespace std;

int main(){

int t[] = { 1, 1, 2, 2, 3, 3, 4, 4, 5, 5 }; listv(t, t+10); set s1(v.begin(),v.end()); if (s1.count(3) == 2) {

s1.erase(3);

}

for(set::iterator i=s1.begin();i!= s1.end(); i++) { cout<<*i<<" ";

} return 0; }
```

- A. program outputs: 1 2 3 4 5
- B. program outputs: 1 2 4 5

C. program outputs: 1 1 2 2 3 4 4 5 5

D. program outputs: 1 1 2 3 3 4 4 5 5

E. compilation error

**ANSWER: A**

### QUESTION NO: 11

Which changes introduced independently will allow the code to compile and display “one” “eight” “nine” “ten”? Choose all that apply.

```
#include
```

```
#include #include using namespace std;
```

```
class A { int a; public:
```

```
A(int a):a(a){}
```

```
int getA() const { return a;}
```

```
/* Insert Code Here 1 */
```

```
};
```

```
/* Insert Code Here 2 */ int main(){
```

```
int t[] = { 3, 4, 2, 1, 6, 5, 7, 9, 8, 10 };
```

```
string s[] = {"three", "four", "two", "one", "six", "five", "seven", "nine", "eight", "ten"}; multimap m; /* Replace Code Here 3 */  
for(int i=0; i<10; i++) {
```

```
m.insert(pair(A(t[i]),s[i]));
```

```
}
```

```
m.erase(m.lower_bound(2),m.upper_bound(7));
```

```
multimap::iterator i=m.begin(); /* Replace Code Here 4 */ for( ; i!= m.end(); i++) {
```

```
cout<<second<<" ";
```

```
}
```

```
cout<<<>
```

```
return 0; }
```

A. operator int() const { return a;} inserted at Place 1

B. bool operator < (const A & b) const { return a

C. bool operator < (const A & b) const { return b.a

D. struct R { bool operator()(const A & a, const A & b) { return a.getA()  
replacing line marked 3 with multimap m; replacong line marked 4 with multimap::iterator i=m.begin();

**ANSWER: A B D**

#### QUESTION NO: 12

Which keywords can be used to define template type parameters? Choose all possible answers:

- A. class
- B. typedef
- C. typename
- D. static
- E. volatile

**ANSWER: A C**

#### QUESTION NO: 13

What happens when you attempt to compile and run the following code? Choose all possible answers.

```
#include  
  
using namespace std;  
  
class C {  
public: int _c; C():_c(0){}  
C(int c) { _c = c;}  
C operator+=(C & b) { C tmp; tmp._c = _c+b._c; return tmp;  
}};  
  
ostream & operator<<(ostream & c, const C & v) { c<<  
template class A { T _v; public:  
A() {}  
A(T v): _v(v){} T getV() { return _v; }  
void add(T & a) { _v+=a; }  
};
```

```
int main()
```

```
{
```

```
A b(2); A a (5); a.add(C());
```

```
cout << a.getV() <
```

**A.** program will display:5

**B.** program will not compile

**C.** program will compile

**D.** program will cause runtime exception

**ANSWER: A C**

#### QUESTION NO: 14

What will happen when you attempt to compile and run the following code?

```
#include
```

```
#include #include
```

```
using namespace std; int main ()
```

```
{
```

```
vectorv1; deque d1;
```

```
for(int i=0; i<5; i++)
```

```
{
```

```
v1.push_back(i);v1.push_front(i);
```

```
d1.push_back(i);d1.push_front(i);
```

```
}
```

```
for(int i=0; i
```

```
{
```

```
cout<
```

```
}
```

```
cout<
```

```
}
```

What will be its output:

- A. 4 4 3 3 2 2 1 1 0 0 0 0 1 1 2 2 3 3 4 4
- B. runtime exception
- C. compilation error due to line 11
- D. compilation error due to line 12

**ANSWER: C**

#### QUESTION NO: 15

What happens when you attempt to compile and run the following code?

```
#include  
  
#include <iostream> using namespace std; class B { int val; public:  
B(int v=0):val(v){} int getV() const {return val;} operator int () const { return val; } };  
ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out; }  
struct Out { ostream & out;  
Out(ostream & o): out(o){}  
void operator()(const T & val ) { out<<val<<endl; }  
B operator()(B & a, B & b) { return a+b; } }; int main() {  
B t[]={1,2,3,4,5,6,7,8,9,10}; vector<B> v1(t, t+10); vector<B> v2(10);  
transform(v1.begin(), v1.end(), v2.begin(), bind2nd(Add(),1)); for_each(v2.rbegin(), v2.rend(), Out(cout));cout<<endl; }  
#endif
```

Program outputs:

- A. 1 2 3 4 5 6 7 8 9 10
- B. 2 3 4 5 6 7 8 9 10 11
- C. 10 9 8 7 6 5 4 3 2 1
- D. 11 10 9 8 7 6 5 4 3 2
- E. compilation error

**ANSWER: E**