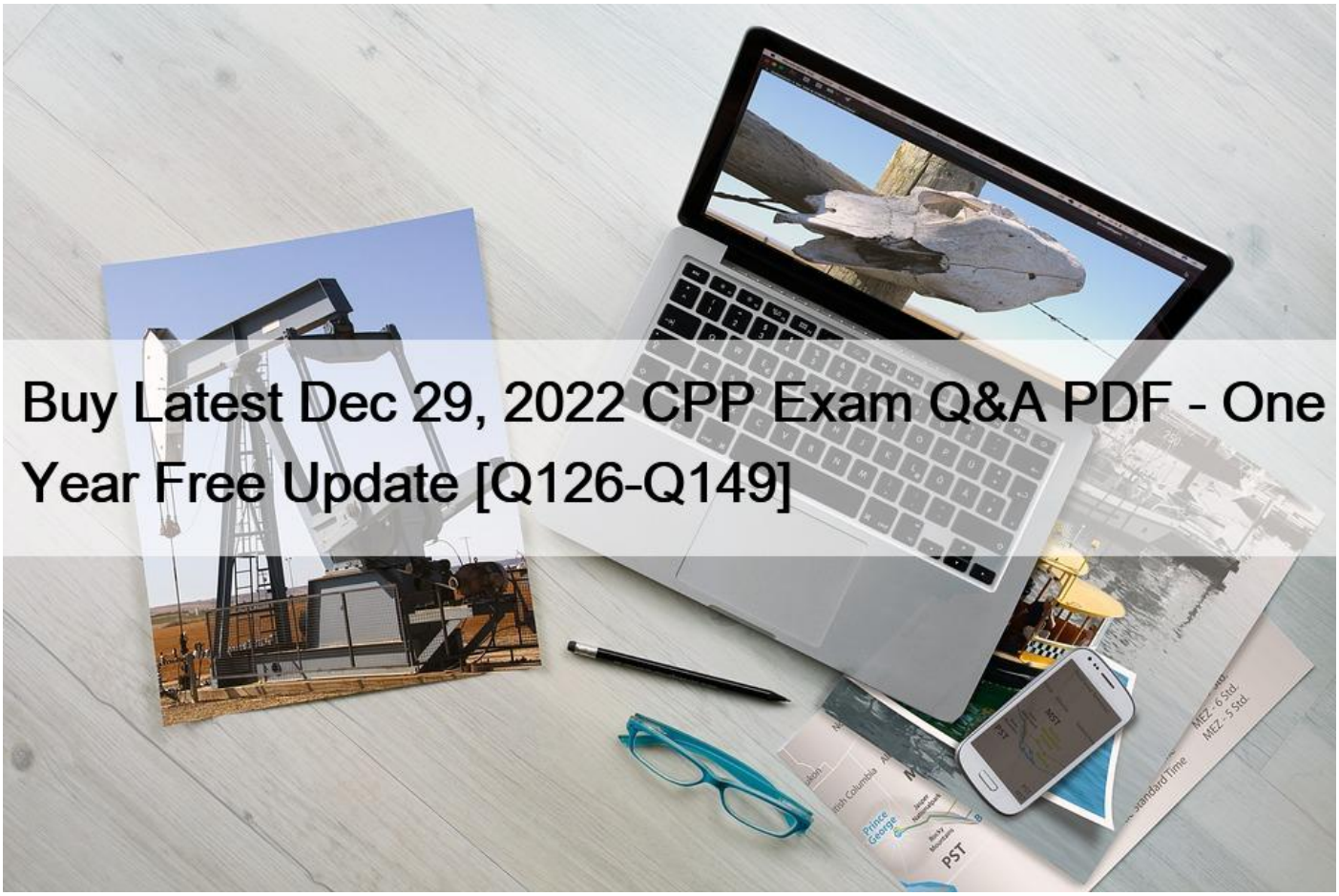
Buy Latest Dec 29, 2022 CPP Exam Q&A PDF - One Year Free Update [Q126-Q149



Buy Latest Dec 29, 2022 CPP Exam Q&A PDF - One Year Free Update

Download the Latest CPP Dump - 2022 CPP Exam Questions NO.126 What happens when you attempt to compile and run the following code?

```
# include <iostream>

# include <set>

# include <vector>

using namespace std;

int main(){

int myints[] ={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };

vector<int>v(myints, myints+10);
```

```
set<int> s1(v.begin(),v.end());

s1.insert(v.begin(),v.end());

s1.erase(s1.lower_bound(2),s1.upper_bound(7));

for(set<int>::iterator i=s1.begin();i!= s1.end(); i++) {

cout<<*i<<&#8221; &#8220;;

}

return 0;

}
```

* program outputs: 0 1 8 9
* program outputs: 2 3 4 5 6 7
* program outputs: 1 6 5 7
* program outputs: 3 4 9 8 0

NO.127 What will happen when you attempt to compile and run the following code?

```
# include <iostream>

# include <set>

# include <vector>

using namespace std;

int main(){

int t[] ={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };

vector<int>v(t, t+10);

set<int> s1(v.begin(),v.end());

s1.insert(v.begin(),v.end());

pair<set<int>::iterator,set<int>::iterator> range;

range = s1.equal_range(6);

cout<<*range.first<<&#8221; &#8220;<<*range.second<<endl;

return 0;

}
```

The output will be:
* 6 6
* 5 7
* 6 7
* 1 5
* 6 5

NO.128 What happens when you attempt to compile and run the following code?

```
# include <iostream>

# include <deque>

# include <list>

# include <stack>

# include <vector>

using namespace std;

int main()

{

deque<int> mydeck;list<int> mylist; vector<int> myvector;

stack<int> first;

stack<int> second(mydeck);

stack<int> third(second);

stack<int, list<int> > fourth(mylist);

fourth.push(10);fourth.push(11);fourth.push(12);

stack<int, vector<int> > fifth(myvector);

fifth.push(10);fifth.push(11);fifth.push(12);

while(!fifth.empty())

{

cout<<fifth.top()<<&#8221; &#8220;;

fifth.pop();
```

}

while (!fourth.empty())

{

cout << fourth.front() << ” “;

fourth.pop();

}

return 0;

}
* program outputs: 12 11 10 12 11 10
* compilation error
* program outputs: 10 11 12 10 11 12
* runtime exception

NO.129 What happens when you attempt to compile and run the following code?

#include <deque>

#include <iostream>

#include <algorithm>

#include <set>

using namespace std;

template<class T>struct Out {

ostream & out;

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

void operator() (const T & val) { out<<val<<” “; } };

int main() {

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

int t1[]={1,2,3,4};

deque<int> d1(t, t+10);

set<int> s1(t, t+10);

```
sort(d1.begin(), d1.end());

cout<<includes(s1.begin(),s1.end(), t1,t1+4)<<&#8221; &#8220;<<includes(d1.begin(),d1.end(), t1,t1+4)

<<endl;

return 0;

}
```

Program outputs:
* 1 1
* 1 0
* 0 1
* 0 0

NO.130 What happens when you attempt to compile and run the following code?

```
# include <iostream>

# include <iomanip>

using namespace std;

int main ()

{

float f = 10.126;

cout<<f<<&#8221; &#8220;<<setprecision(2)<<f<<endl;

return 0;

}
```

Program outputs:
* 10.126 10
* 10.126 10.12
* compilation error
* 10.126 10.13

NO.131 What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>
```

```
using namespace std;

void myfunction(int i) {

cout << &#8221; &#8221; << i;

}

int main() {

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

vector<int> v1(t, t + 10);

copy_backward(t, t+10, v1.rend());

for_each(v1.begin(), v1.end(), myfunction);

return 0;

}
```

Program outputs:
* 10 5 9 6 2 4 7 8 3 1
* 1 3 8 7 4 2 6 9 5 10 10 5 9 6 2 4 7 8 3 1
* 1 3 8 7 4 2 6 9 5 10
* runtime exception/segmentation fault
* compilation error

NO.132 What happens when you attempt to compile and run the following code?

```
#include <iostream>

using namespace std;

int main()

{

cout.setf(ios::oct, ios::basefield);

cout<<100<<&#8221; &#8220;;

cout.setf(ios::showbase);

cout<<100<<&#8221; &#8220;;

return 0;

}
```

Program outputs:
* 144 0144
* 144 0x64
* 0x144 0144
* 0144 100
* compilation error

NO.133 What happens when you attempt to compile and run the following code?

#include <iostream>

#include <map>

#include <vector>

#include <string>

using namespace std;

int main(){

int second[] ={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };

string first[] = {“three”, “four”, “two”, “one”, “six”,”five”, “seven”, “nine”,”eight”,”zero”}; map<int,string> m; for(int i=0; i<10; i++) { m.insert(pair<int,string>(second[i],first[i]));

}

m[0]=”ten”;

m.insert(pair<int,string>(1,”eleven”));

for(map<int, string>::iterator i=m.begin();i!= m.end(); i++) {

cout<<i?>second<<” “;

}

return 0;

}
* program outputs: zero one two three four five six seven eight nine
* program outputs: ten one two three four five six seven eight nine
* program outputs: zero eleven two three four five six seven eight nine
* program outputs: ten eleven two three four five six seven eight nine
* program outputs: 0 1 2 3 4 5 6 7 8 9

NO.134 What happens when you attempt to compile and run the following code?

```
#include <vector>

#include <iostream>

int main ()

{

int t[]={1,2,3,4,5};

std::vector<int>v1(t,t+5);

std::vector<int>v2(v1);

v1.resize(10);

v2.reserve(10);

std::vector<int>::iterator i = v1.begin();int ii = 0;

while (i != v1.end()) { std::cout<<i[ii]<<&amp;#8221; &amp;#8220;;ii??;i++; }

i = v2.begin();ii=0;

while (i != v2.end()) { std::cout<<i[ii]<<&amp;#8221; &amp;#8220;;ii??;i++; }

return 0;

}
```

* program outputs 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
* compilation error
* program outputs 1 1 1 1 1 1 1 1 1 1 1 2 3 4 5
* program outputs 1 2 3 4 5 0 0 0 0 0 1 2 3 4 5 0 0 0 0 0

NO.135 Which are NOT valid instantiations of priority_queue object:

```
# include <iostream>

# include <deque>

# include <list>

# include <queue>

# include <vector>

using namespace std;
```

```
int main()

{

deque<int> mydeck;list<int> mylist; vector<int> myvector;

priority_queue<int> first;//line I

priority_queue<int, deque<int> > second;//line II

priority_queue<int> third(first);//line III

priority_queue<int, list<int> > fourth(third);//line IV

priority_queue<int, vector<int> > fifth(myvector.begin(), myvector.end());//line V return 0;

}
```

* line I
* line II
* line III
* line IV
* line V

NO.136 What happens when you attempt to compile and run the following code?

```
# include <iostream>

# include <map>

# include <vector>

# include <string>

using namespace std;

int main(){

int second[] ={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };

string first[] = {&#8220;three&#8221;, &#8220;four&#8221;, &#8220;two&#8221;, &#8220;one&#8221;,
&#8220;six&#8221;,&#8221;five&#8221;, &#8220;seven&#8221;,
&#8220;nine&#8221;,&#8221;eight&#8221;,&#8221;zero&#8221;}; map<int,string> m;

for(int i=0; i<10; i++) {

m.insert(pair<int,string>(second[i],first[i]));

}

m[0]=&#8221;ten&#8221;;
```

m.insert(pair<int,string>(1,”eleven”));

for(map<int, string>::iterator i=m.begin();i!= m.end(); i++) {

cout<<i?>second<<” “;

}

return 0;

}
* program outputs: zero one two three four five six seven eight nine
* program outputs: ten one two three four five six seven eight nine
* program outputs: zero eleven two three four five six seven eight nine
* program outputs: ten eleven two three four five six seven eight nine
* program outputs: 0 1 2 3 4 5 6 7 8 9

NO.137 What happens when you attempt to compile and run the following code?

include <deque>

include <set>

include <iostream>

include <algorithm>

using namespace std;

class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;} bool operator < (const B & v) const { return val<v.val;} }; ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;} template<class T>struct Out {

ostream & out;

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

void operator() (const T & val) { out<<val<<” “; } };

int main() {

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

deque<B> d1(t, t+10);

```
sort(d1.begin(), d1.end());

set<B> s1(t,t+10);

cout<<binary_search(s1.begin(),s1.end(), B(4))<<&#8221; &#8220;<<binary_search(d1.begin(),d1.end(),

B(4))<<endl;

return 0;

}
```

Program outputs:
* 1 0
* 1 1
* 0 0
* 0 1
* compilation error

NO.138 What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <map>

using namespace std;

int main() {

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

map<int, int> m;

for(int i=0; i < 10; i++) {

m[i]=t[i];

}

pair<const int,int> p(5,5);

map<int, int>::iterator it = find(m.begin(), m.end(), p);

if (it != m.end())

{
```

cout<<it?>first<<endl;

}

else

{

cout<<“Not found!n”;

}

return 0;

}

Program outputs:
* 5
* Not found!
* 10
* compilation error

NO.139 What will happen when you attempt to compile and run the following code?

#include <iostream>

using namespace std;

template <typedef T>

class A {

T_v;

public:

A(T v): _v(v){}

T getV() { return _v; }

};

int main()

{

A<int> a(1);

cout << a.getV() <<endl;

```
return 0;

}
```

* program will display:1
* program will not compile
* program will compile
* program will cause runtime exception

NO.140 What will happen when you attempt to compile and run the code below, assuming that you enter the following sequence: true false<enter>?

```
#include <iostream>

#include <string>

using namespace std;

int main ()

{

bool a,b;

cin>>boolalpha>>a>>b;

cout<<a<<b<<endl;

return 0;

}
```

Program will output:
* truefalse
* true0;
* 1false
* 10
* none of these

NO.141 What happens when you attempt to compile and run the following code?

```
#include <iostream>

using namespace std;

int main()

{

cout.setf(ios::hex, ios::basefield);
```

cout<<100<<” “;

cout.flags(ios::showbase);

cout<<100<<” “;

return 0;

}

Program outputs:
* 64 64
* 64 0x64
* 0x64 0x64
* 64 100
* compilation error

NO.142 What happens when you attempt to compile and run the following code?

include <vector>

include <iostream>

include <algorithm>

using namespace std;

class B { int val;

public:

B(int v):val(v){}

int getV() const {return val;} bool operator < (const B & v) const { return val<v.val;} }; ostream & operator <<(ostream & out, const B & v) { out<<v.getV(); return out;} template<class T>struct Out {

ostream & out;

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

void operator() (const T & val) { out<<val<<” “; } };

int main() {

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

vector<B> v1(t, t+10);

sort(v1.begin(), v1.end());

```
for_each(v1.begin(), v1.end(), Out<B>(cout));cout<<endl;

return 0;

}
```

Program outputs:
* 8 10 5 1 4 6 2 7 9 3
* 1 2 3 4 5 6 7 8 9 10
* compilation error
* 10 9 8 7 6 5 4 3 2 1

NO.143 What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

void myfunction(int i) {

cout << &amp;#8221; &amp;#8221; << i;

}

int main() {

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

vector<int> v1(t, t + 10);

copy(t, t+10, v1.end());

for_each(v1.begin(), v1.end(), myfunction);

return 0;

}
```

Program outputs:
* 10 5 9 6 2 4 7 8 3 1
* 10 5 9 6 2 4 7 8 3 1 10 5 9 6 2 4 7 8 3 1
* compilation error
* runtime exception/segmentation fault

NO.144 What happens when you attempt to compile and run the following code?

```
# include <vector>

# include <iostream>

# include <algorithm>

# include <functional>

using namespace std;

template<class T>struct Out {

ostream & out;

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

void operator() (const T & val ) { out<<val<<&amp;#8221; &amp;#8220;; } };

struct Add : public binary_function<int, int, int> {

int operator() (const int & a, const int & b) const {

return a+b;

}

};

int main() {

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

vector<int> v1(t, t+10);

vector<int> v2(10);

transform(v1.begin(), v1.end(), v2.begin(), bind1st(Add(), 1));

for_each(v2.rbegin(), v2.rend(), Out<int>(cout));cout<<endl;

return 0;

}
```

Program outputs:
* 1 2 3 4 5 6 7 8 9 10
* 2 3 4 5 6 7 8 9 10 11
* 10 9 8 7 6 5 4 3 2 1
* 11 10 9 8 7 6 5 4 3 2
* compilation error

NO.145 What happens when you attempt to compile and run the following code?

```
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
template<class T>struct Out {
ostream & out;
Out(ostream & o): out(o){}
void operator()(const T & val ) {
out<<val<<&#8221; &#8220;;
}
};
struct Sequence {
int start;
Sequence(int start):start(start){}
int operator()() { return 10*(1+(start++ %3)); } };
int main() {
vector<int> v1(10);
vector<int> v2(10);
generate(v1.begin(), v1.end(), Sequence(1));
sort(v1.rbegin(), v1.rend());
unique_copy(v1.begin(),v1.end(), v2.begin());
for_each(v2.begin(), v2.end(), Out<int>(cout) );cout<<endl;
return 0;
}
```

Program outputs:
* 20 30 10 20 30 10 20 30 10 20
* 30 20 10 0 0 0 0 0 0 0
* 30 0 0 0 0 0 0 0 20 10
* compilation error

NO.146 What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <set>

#include <vector>

using namespace std;

int main(){

int myints[] ={ 3, 4, 2, 1, 6, 5, 7, 9, 8, 0 };

vector<int>v(myints, myints+10);

set<int> s1(v.begin(),v.end());

set<int, greater<int> > s2(v.begin(), v.end());

for(set<int>::iterator i=s1.begin();i!= s1.end(); i++) {

cout<<*i<<&#8221; &#8220;;

}

for(set<int, greater<int> >::iterator i=s2.begin();i!= s2.end(); i++) { cout<<*i<<&#8221; &#8220;;

}

cout<<endl;

return 0;

}
```

* program outputs: 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
* program outputs: 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
* program outputs: 0 1 2 3 4 5 6 7 8 9 9 8 7 6 5 4 3 2 1 0
* program outputs: 9 8 7 6 5 4 3 2 1 0 0 1 2 3 4 5 6 7 8 9

Explanation/Reference:

NO.147 What happens when you attempt to compile and run the following code?

```
#include <iostream>

#include <algorithm>

#include <vector>

# include <deque>

# include <set>

using namespace std;

void myfunction(int i) {

cout << &amp;#8221; &amp;#8221; << i;

}

int main() {

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

vector<int> v1(t, t + 10);

deque<int> d1(t, t + 10);

set<int> s1(t, t + 10);

for_each(v1.begin(), v1.end(), myfunction); // Line I

for_each(d1.begin(), d1.end(), myfunction); // Line II

for_each(s1.begin(), s1.end(), myfunction); // Line III

return 0;

}
```

* program outputs: 10 5 9 6 2 4 7 8 3 1 10 5 9 6 2 4 7 8 3 1 1 2 3 4 5 6 7 8 9 10
* program outputs: 10 5 9 6 2 4 7 8 3 1 10 5 9 6 2 4 7 8 3 1 10 5 9 6 2 4 7 8 3 1
* program outputs: 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10
* compilation error in line I
* compilation error in line III

NO.148 What happens when you attempt to compile and run the following code?

```
# include <deque>

# include <iostream>

# include <algorithm>
```

using namespace std;

template<class T>struct Out {

ostream & out;

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

void operator() (const T & val) { out<<val<<” “; } };

int main() {

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

deque<int> d1(t, t+10);

sort(d1.begin(), d1.end());

deque<int>::iterator it = upper_bound(d1.begin(), d1.end(), 4);

for_each(it, d1.end(), Out<int>(cout));cout<<endl;

return 0;

}

Program outputs:
* 5 6 7 8 9 10
* 4 5 6 7 8 9 10
* 1 2 3 4 5 6 7 8 9 10
* 1 2 3 4 5
* 1 2 3 4

NO.149 What happens when you attempt to compile and run the following code?

#include <deque>

#include <iostream>

#include <algorithm>

#include <set>

using namespace std;

template<class T>struct Out {

ostream & out;

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

void operator() (const T & val) { out<<val<<” “; }

};

bool Compare(char a, char b) { return tolower(a) < tolower(b);}

int main() {

char s[]={“qwerty”};

char t1[]={“ert”};

char t2[]={“ERT”};

sort(s, s+6);

cout<<includes(s,s+6, t1,t1+3, Compare)<<” “;<<includes(s,s+6, t2,t2+3, Compare)<<endl; return 0;

}

Program outputs:
* 0 0
* 0 1
* 1 0
* 1 1

Verified CPP Dumps Q&As - 1 Year Free & Quickly Updates: https://www.dumpsmaterials.com/CPP-real-torrent.html]