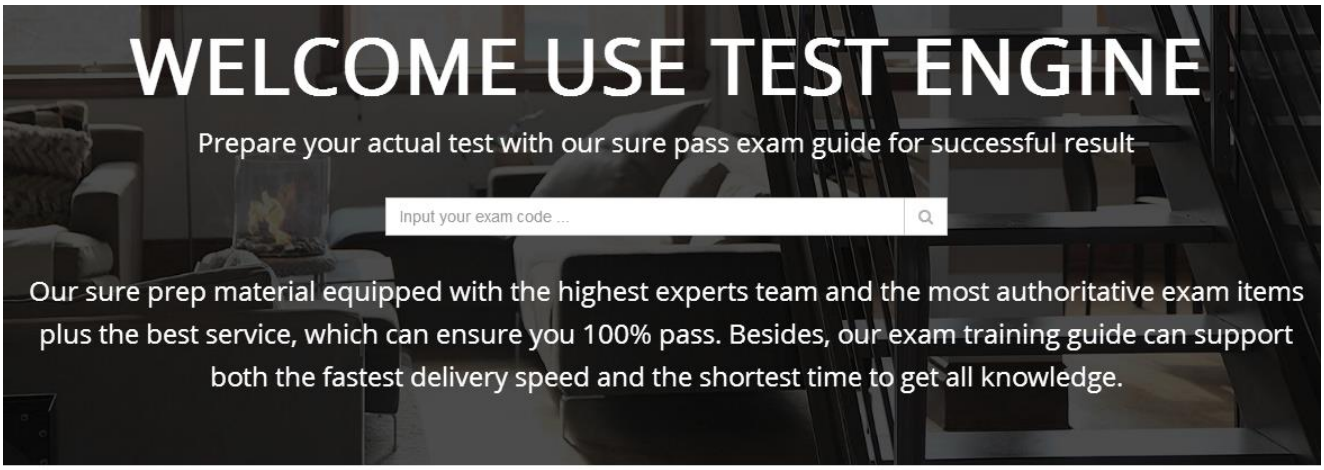
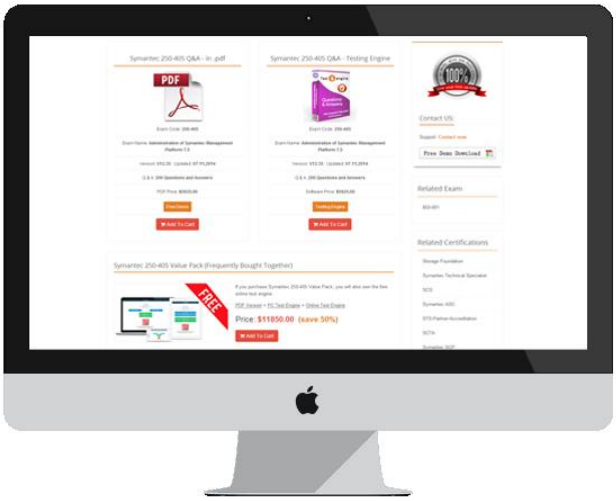
Prep4sureGuide

# Prep4sureGuide



### Quality and Value

Prep4sureGuide Practice Exams are written to the highest standards of technical accuracy, using only certified subject matter experts and published authors for development - no all dumps.

### Tested and Approved

We are committed to the process of vendor and third party approvals. We believe professionals and executives alike deserve the confidence of quality coverage these authorizations provide.

### Easy to Pass

If you prepare for the exams using our Prep4sureGuide testing engine, It is easy to succeed for all certifications in the first attempt. You don't have to deal with all dumps or any free torrent / rapidshare all stuff.

### Try Before Buy

Prep4sureGuide offers free demo of each product. You can check out the interface, question quality and usability of our practice exams before you decide to buy.

| HAPPY CUSTOMERS | DOWNLOADS | TEAM MEMBERS | SHARES |
|---|---|---|---|
| 32694 | 62152 | 32694 | 56692 |

http://www.prep4sureguide.com

Prepare your actual test with our sure pass exam guide for successful result

**Exam** : **CPA-21-02**

**Title** : CPA - C++ Certified Associate Programmer

**Vendor** : C++ Institute

**Version** : DEMO

**NO.1** What will happen when you attempt to compile and run the following code?

```
#include <iostream>
using namespace std;
int getValue();
int main()
{
const int x = getValue();
cout<<x;
return 0;
}
int getValue()
{
return 5;
}
```

**A.** It will print 0

**B.** The code will not compile.

**C.** It will print 5

**D.** It will print garbage value

***Answer:*** C

**NO.2** What happens when you attempt to compile and run the following code?

```
#include <iostream>
using namespace std;
int main()
{
int *t;
t = new int[2];
for (int i=0; i<2; i++) {
t[i]=0;
}
cout << t[1];
}
```

**A.** It prints: 0

**B.** It prints: 1

**C.** It prints: 2

**D.** It prints: 3

***Answer:*** A

**NO.3** What happens when you attempt to compile and run the following code?

```
#include <iostream>
#include <string>
using namespace std;
class A {
protected:
int y;
```

```
public:
int x;
int z;
A() { x=1; y=2; z=3; }
A(int a, int b) : x(a), y(b) { z = x * y;}
void Print() {
cout << z;
}
};
int main () {
A a(2,5);
a.Print();
return 0;
}
```

**A.** It prints: 10
**B.** It prints: 2
**C.** It prints: 6
**D.** It prints: 5
***Answer:*** A

**NO.4** Which of the structures is incorrect?

```
1:
struct s1{
int x;
long int li;
};
2:
struct s2{
float f;
struct s2 *s;
};
3:
struct s3{
float f;
struct s3 s;
};
```

**A.** 1
**B.** 2
**C.** 3
**D.** 2, 3
***Answer:*** C

**NO.5** Which code lines inserted independently instead of the comment will make the following program work correctly? (Choose three.)

```
#include <iostream>
using namespace std;
//Insert line of code here
{
      cout << "Hello, it's me" << endl;
}
```

**A.** int main (int argc, char *argv[])
**B.** int main (int c, char *v[])
**C.** int main
**D.** void main ()
***Answer:*** A,B

**NO.6** What is the output of the program?
#include <iostream>
#include <string>
using namespace std;
int main () {
string s1 = "Hello", s2 = "World";
s2 = s1 + s2;
cout << s2;
return 0;
}
**A.** It prints: Hello
**B.** It prints: HelloWorld
**C.** It prints: WorldHello
**D.** It prints: WorldHelloWorld
***Answer:*** B

**NO.7** What is the output of the program?
#include <iostream>
using namespace std;
class BaseC
{
int i;
public:
BaseC() { i=?1;}
BaseC(int i) { i=i; }
void seti(int a) { i = a; };
void Print() { cout << i; }
};
int main()
{
BaseC *o = new BaseC();

```
o?>seti(10);
o?>Print();
}
```

**A.** It prints: 10
**B.** It prints: ?1
**C.** It prints: 0
**D.** Compilation error
***Answer:*** A

**NO.8** What happens when you attempt to compile and run the following code?

```
#include <iostream>
using namespace std;
class BaseC
{
public:
int *ptr;
BaseC() { ptr = new int(10);}
BaseC(int i) { ptr = new int(i); }
~BaseC() { delete ptr; }
};
void fun(BaseC x);
int main()
{
BaseC *o = new BaseC(5);
fun(*o);
}
void fun(BaseC x) {
cout << "Hello:"<<*x.ptr;
}
```

**A.** It prints: Hello:50
**B.** It prints: Hello:10
**C.** It prints: Hello:5
**D.** Compilation error
***Answer:*** C

**NO.9** What happens when you attempt to compile and run the following code?

```
#include <iostream>

using namespace std;

class Base {
public:
     Base() {
          cout << "A";
     }
     ~Base() {
          cout << "D";
     }
     Base(Base& b) {
          cout << "C";
     }
     Base& operator=(Base &b) {
          cout << "B";
          return *this;
     }
};
int main()
{
     Base b1, b2, *b3 = new Base();
     b2 = *b3;
}
```

**A.** It prints: AAABDD
**B.** It pints: AABD
**C.** It prints: AABDD
**D.** It causes a compilation error
***Answer:*** D

**NO.10** What happens when you attempt to compile and run the following code?
#include <iostream>
using namespace std;
int main()
{
const char *s;

```
char str[] = "Hello";
s = str;
while(*s) {
cout << *s++;
}
return 0;
}
```

**A.** It prints: el
**B.** It prints: Hello
**C.** It prints: H
**D.** It prints: o
***Answer:*** B

**NO.11** What is the output of the program?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
string s1="World";
string s2;
s2="Hello" + s1;
cout << s2;
return( 0 );
}
```

**A.** It prints: HelloWorld
**B.** It prints: Hello
**C.** It prints: World
**D.** Compilation error
***Answer:*** A

**NO.12** What happens when you attempt to compile and run the following code?

```
#include <iostream>
using namespace std;
int main(){
int *i;
i = new int;
*i = 1.0 / 2 * 2 / 1 * 2 / 4 * 4;
cout << *i;
return 0;
}
```

**A.** It prints: 0
**B.** It prints: 1
**C.** It prints: 2
**D.** It prints: 0.5

***Answer:*** C

**NO.13** What happens when you attempt to compile and run the following code?

```
#include <iostream>

using namespace std;

class Test {
     float i, j;
};

class Add {
public:
     int x,y;
     Add (int a=3, int b=3) { x=a; y=b; }
     int result() {return x+y; }
};

int main()
{
     Test test;
     Add* padd;
     padd = &test;
     cout << padd -> result();
}
```

**A.** It prints: 33
**B.** It prints: 007
**C.** It causes a compilation error
**D.** lit prints: 9
***Answer:*** C

**NO.14** Which of the following structures are correct?
1:
struct s1{
int x;
char c;
};
2:

```
struct s2{
float f;
struct s2 *s;
};
3:
struct s3{
float f;
in i;
}
```

**A.** 1
**B.** 2
**C.** 3
**D.** All of these
***Answer:*** A,B

**NO.15** What happens when you attempt to compile and run the following code?

```
#include <iostream>
#include <string>
using namespace std;
struct Person {
string name;
int age;
};
class First
{
Person *person;
public:
First() {person = new Person;
person?>name = "John";
person?>age = 30;
}
void Print(){
cout<<person?>name << " "<< person?>age;
}
};
int main()
{
First t[2];
for (int i=0; i<2; i++)
t[i].Print();
}
```

**A.** It prints: 30
**B.** It prints: John
**C.** It prints: John 31
**D.** It prints: John 30John 30

***Answer:*** D

**NO.16** What happens when you attempt to compile and run the following code?

```
#include <iostream>
using namespace std;
int f(int a, int b);
int main()
{
float b;
b = f(20,10);
cout << b;
return 0;
}
int f(int a, int b)
{
return a/b;
}
```

**A.** It prints: 2
**B.** It prints: 5
**C.** It prints: 10
**D.** It prints: 0
***Answer:*** A

**NO.17** What happens when you attempt to compile and run the following code?

```
#include <iostream>
#include <string>

using namespace std;

void print(string c);

int main()
{
     print("Test");
}
void print(string c)
{
     cout << c;
}
```

**A.** It prints: T

**B.** It prints an empty line
**C.** It prints: Tesc
**D.** It prints: st
***Answer:*** C

**NO.18** What happens when you attempt to compile and run the following code?

```
#include <iostream>

using namespace std;

int main()
{
      const char* s = "12";
      cout << s + 1;
}
```

**A.** It prints: 13
**B.** It prints: l
**C.** It prints: 12
**D.** It prints: 2
***Answer:*** D

**NO.19** What happens when you attempt to compile and run the following code?

```
#include <iostream>
#include <exception>

using namespace std;

class myClass : public exception
     virtual const char* what() const throw()
     {
          return "My exception.";
     }
}obj;

int main()
{
     try {
          throw obj;
     }
     catch (exception& e) {
          cout << e.what() << endl;

     }
}
```

**A.** It prints: 1
**B.** It causes a compilation error
**C.** It prints: 0
**D.** It prints: My exception,
***Answer:*** B

**NO.20** Which code, inserted at line 14, generates the output "3.14 10"?
#include <iostream>
using namespace std;
namespace myNamespace1
{
int x = 5;
int y = 10;
}
namespace myNamespace2
{

```
float x = 3.14;
float y = 1.5;
}
int main () {
//insert code here
cout << x << " " << y;
return 0;
}
```

**A.** using myNamespace2::y; using myNamespace1::x;

**B.** using namespace myNamespace1;

**C.** using namespace myNamespace1; using namespace myNamespace2;

**D.** using myNamespace1::y; using myNamespace2::x;

***Answer:*** D

**NO.21** What happens when you attempt to compile and run the following code?

```
#include <iostream>
using namespace std;
int main() {
int i, j;
for(i = 0; i < 2; i++) {
for(j = i; j < i + 1; j++)
if(j == i)
continue;
else
break;
}
cout << j;
return 0;
}
```

**A.** It prints: 0

**B.** It prints: 3

**C.** It prints: 2

**D.** It prints: 1

***Answer:*** C

**NO.22** What happens when you attempt to compile and run the following code?

```
#include <iostream>
using namespace std;
int compare(int, int);
int main()
{
int x = compare(10, 20);
cout << x;
return 0;
}
```

```
int compare(int i, int j)
{
return i<j;
}
```

**A.** It prints: 0
**B.** It prints: 2
**C.** It prints: 1
**D.** It prints: 10
***Answer:*** C

**NO.23** What happens when you attempt to compile and run the following code?

```
#include <iostream>
using namespace std;
int op(int x, int y);
int main()
{
int i=2, j=2, k;
float f=0.3;
k = op(i, j);
cout<< k << "," << op(1, f);
return 0;
}
int op(int x, int y)
{
return x+y;
}
```

**A.** It prints: 4,1
**B.** It prints: 4,0.7
**C.** It prints: 4,0
**D.** It prints: 0,4
***Answer:*** A

**NO.24** What happens when you attempt to compile and run the following code?

```
#include <iostream>
using namespace std;
int min(int a, int b);
int main()
{
int b=10;
b = min(5,20);
cout << b;
return 0;
}
int min(int a, int b)
{
```

```
if (a<b)
return(a);
else
return(b);
}
```

**A.** It prints: 10
**B.** It prints: 20
**C.** It prints: 5
**D.** It prints: 0
***Answer:*** C

**NO.25** Which code, inserted at line 10, generate the output "50"?

```
#include <iostream>
using namespace std;
class Base {
int age;
public:
Base () {
age=5;
};
//insert code here
void Print() { cout << age;}
};
void setAge(Base &ob) {ob.age = 0;}
int main () {
Base a;
a.Print();
setAge(a);
a.Print();
return 0;
}
```

**A.** friend void setAge(Base ob);
**B.** friend void setAge(Base *ob);
**C.** friend void setAge(Base &ob);
**D.** None of these
***Answer:*** C

**NO.26** What happens when you attempt to compile and run the following code?

```
#include <iostream>
#include <exception>

using namespace std;

class myClass : public exception
{
     virtual const char* what() const throw()
     {
          return "A";
     }
}obj;

int main()
{
     try {
          cout << "B";
          throw obj;
          cout << "C";
     }
     catch (exception& e) {
          cout << e.what();

     }
     cout << "D";
}
```

**A.** It prints: BAD
**B.** It prints: BACD
**C.** It prints: ABCD
**D.** It prints: BAC
***Answer:*** B