C++ Quiz

Please select the best answer to the questions below. Some questions may have more than one answer. You have 30 minutes to complete.

Question 1

A function call is always an rvalue. True or false?

1. True
2. False

Question 2

What is the difference between a class and a structure?

1. A class can have virtual functions, structure cannot.
2. A class can be derived from another class, structure cannot be derived from another structure.
3. In class, all members are private by default whereas in structure, they are public.
4. Operators can be overloaded for a class, not for a structure.
5. All of the above.

Question 3

```
namespace NS {
  void F() {
    x++;
  }
  int x;
}
```

In the namespace NS, the function F can access the variable x also declared in namespace NS?

1. True
2. False

Question 4

```
#include <iostream>
#include <deque>
using namespace std;
int main() {
  deque<int> d;
  d.push_back(1);
  d.push_back(2);
  d.push_back(3);
  d.push_back(4);
  for (deque<int>::iterator i = d.begin(); i != d.end(); i++)
    if ((*i) % 2 == 0)
      d.erase(i);
  for (deque<int>::iterator i = d.begin(); i != d.end(); i++)
    cout << (*i) << endl;
}
```

What does the program above do?

1. Remove all even numbers from the deque.
2. Remove all odd numbers from the deque.
3. Remove all numbers from the deque.
4. This program will not compile.
5. Undefined or compiler dependent result, will probably result in segmentation fault.

Question 5

```
class Base {
public:
  ~Base(){} //(A)
};
class Derived: public Base {
public:
  ~Derived(){} //(B)
};
int main() {
  Base * b = new Derived();
  delete b;
}
```

In which order are the destructors called?
1. A and then B
2. B and then A
3. A only
4. B only
5. Won't compile

Question 6

```
template <typename T> class FOO {
public:
  int B(T t); //(A)
  int B(int i); //(B)
  int B(int i) const; //(C)
};
template<> class FOO<bool> {
public:
  int B(int i); //(D)
  int B(int i) const; //(E)
};
int main() {
  FOO<bool> f;
  const FOO<bool> g = f;
  g.B(10); //(F)
}
```

Which function gets called at (F) ?
1. A
2. B
3. C
4. D
5. E

Question 7

In addition to c-style, which casts can be used to cast an int to a pointer or a pointer to an int?
1. static_cast
2. dynamic_cast
3. reinterpret_cast
4. pointer_cast
5. None of the above

Question 8

```
#include <iostream>
struct Base {
    Base() { std::cout << "Hello World" << std::endl; }
    unsigned int color;
};
struct C1 : virtual public Base { };
struct C2 : virtual public Base { };
struct C3 : public Base { };
struct C4 : public Base { };
struct Derived : public C1, public C2, public C3, public C4 { };

int main() {
    Derived d;
    return 0;
}
```

How many times is "Hello World" displayed on output?

1. 1
2. 2
3. 3
4. Compilation error
5. Compiler dependent

Question 9

```
void AllocateInteger(int* p) {
  p = new int;
  *p++ = 2;
}
int main() {
  int i = 1;
  AllocateInteger(&i);
  std::cout<<i;
  return 0;
}
```

What value is displayed on output?

1. Outputs '1'
2. Outputs '2'
3. Outputs '3'
4. Output is undefined
5. Will not compile or compiler dependent

Question 10

What does the following function do?

```
int f(int a, int b) {
  return a % b ? f(b, a % b) : b;
}
```

1. Compute least common denominator
2. Compute aCb ("a" combination "b")
3. Divides larger number by smaller number and returns remainder 4. Compute polynomial coefficient 5. None of the above

Question 11

Which of the following operators cannot be overloaded?

1. +=
2. []
3. ,
4. =
5. All operators mentioned above can be overloaded.

Question 12
You cannot overload left shift (<<) and right shift (>>) operators for classes because they are already defined in iostream.
1. True
2. False

Question 13

```
#include <iostream>
int main() {
  int x = 3;
  switch(x) {
    case 0:
      int x = 1;
      std::cout << x << std::endl;
      break;
    case 3:
      std::cout << x << std::endl;
      break;
    default:
      x = 2;
      std::cout << x << std::endl;
  }
  return 0;
}
```

1. 0
2. 1
3. 2
4. 3
5. Ill-formed

Question 14

```
#include <iostream>
struct Car {
  Car() : price(20000) {}
  Car(double b) : price(b*1.1) {}
  double price;
};

struct Toyota : public virtual Car {
  Toyota(double b) : Car(b) {}
};

struct Prius : public Toyota {
  Prius(double b) : Toyota(b) {}
```

```
};

int main(int argc, char** argv) {
  Prius p(30000);
  std::cout << p.price << std::endl;
  return 0;
}
```

1. 20000
2. 22000
3. 30000
4. 33000
5. Code is ill-formed

Question 15

In C++, a class can have 2 functions with identical list of parameters but different return types.

1. True
2. False

C++ Test Answers

Answers
1. 2
2. 3
3. 2
4. 5
5. 3
6. 5
7. 3
8. 3
9. 1
10. 5
11. 5
12. 2
13. 5
14. 1
15. 2