

Sample Test of C++

brian m. carlson

This test omits the relevant headers for code snippets; it is assumed that you are familiar with which headers are necessary for certain functions.

- (1) The following code produces two lines. What are they?

```
int main(void)
{
    std::cout << std::hex << 5 << std::endl;
    std::cout << 105 << std::endl;

    return 0;
}
```

- (2) This code fails to compile with an error on the indicated line. How do you need to adjust the code so it compiles and operates correctly.

```
class A
{
    public:
        A()
        {
            a = 5;
            b = 3;
        }
    protected:
        virtual int function()
        {
            c = a + b;
            a++;
            return c;
        }
        int a, b, c;
};

class B : public A
{
};

class C : public A
{
};

class D : public B, public C
{
    public:
        virtual int method()
        {
            return A::function(); // error
        }
};
```

```

int main(void)
{
    D obj;

    std::cout << obj.method() << std::endl;
    std::cout << (obj.method() * 2) << std::endl;

    return 0;
}

```

(3) Write a function to efficiently exchange the contents of two `std::vectors`.

(4) What does the following statement do?

```
using namespace std;
```

(5) Determine whether this program runs successfully (exits 0) for 32-bit and 64-bit Windows and 32-bit and 64-bit Unix systems. Provide the exit value if it does not.

```

template<class T>
int GetValue()
{
    return sizeof(T) * 8;
}

```

```

int main(void)
{
    return GetValue<long>() - GetValue<int>();
}

```

(6) Provide the three lines of output.

```

int GetByte(int &t, int &u, int *s)
{
    t++;
    int &x = s[t];
    u += s[t];
    int &y = s[u];
    std::swap(x, y);
    return s[(x + y) & 15];
}

```

```

int main(void)
{
    int s[16], t = 0, u = 2;

    for (int i = 0; i < 16; i++)
        s[i] = i;

    std::cout << GetByte(t, u, s) << std::endl;
    std::cout << GetByte(t, u, s) << std::endl;
    std::cout << GetByte(t, u, s) << std::endl;

    return 0;
}

```