Instructions

For a each topic (if applicable) there are 4 sections:

Fix the Broken Code

Modify the code in the cleanest (and preferably shortest) way possible, so that it will compile and run without errors or warnings.

Trace the Working Code

Trace through the execution of the code, keeping track of the value of each variable at each point in time, and the final output that the program produces.

Circle the Variables' Scopes

Circle the scope of the given variable – i.e., circle the part of the code in which the variable "exists". A good test, if you're not sure, is: if you inserted a statement to `cout` the variable at a given point in the code, would any errors be produced?

Fill in the Memory

Fill in (or draw) a picture representing the computer memory, showing how the variables might be allocated, and whether they have an assigned value, or are uninitialized (use `?` for "uninitialized").

1 Variables and Assignment

Fix the Broken Code

```
#include <iostream>
using namespace std;

int main() {
    int a = 5;
    int b = 7;

    cout << "before swapping: " << a << " " << b << endl;

    // swap
    int temp = a;
    a = b;
    b = temp;

    cout << "after swapping: " << a << " " << b << endl;

    return 0;
}
```

```
--- code/1-fix-1.cpp        2014-04-30 22:09:04.000000000 -0700
+++ code/1-fix-1.answer.cpp        2014-04-30 22:26:24.000000000 -0700
@@ -8,9 +8,9 @@
     cout << "before swapping: " << a << " " << b << endl;

     // swap
-    int temp = int a;
-    int a = int b;
-    int b = int temp;
+    int temp = a;
+    a = b;
+    b = temp;

     cout << "after swapping: " << a << " " << b << endl;
```

```
before swapping: 5 7
after swapping: 7 5
```

```cpp
#include <iostream>
#include <string>
using namespace std;

int main() {
    int    i = '5' + 4.7;  // '5' == 53
    char   c = 42;         // '*' == 42
    bool   b = 10;
    double d =  7;
    string s = "hello world!";

    int u = 42;  // otherwise, we're using an uninitialized variable below

    cout << i << " " << c << " " << b << " "
         << d << " " << s << " " << u << endl;

    return 0;
}
```

```
--- code/1-fix-2.cpp        2014-04-30 22:21:38.000000000 -0700
+++ code/1-fix-2.answer.cpp        2014-04-30 22:27:01.000000000 -0700
@@ -9,7 +9,7 @@
     double d =  7;
     string s = "hello world!";

-    int u;
+    int u = 42;  // otherwise, we're using an uninitialized variable below

     cout << i << " " << c << " " << b << " "
          << d << " " << s << " " << u << endl;
```

```
57 * 1 7 hello world! 42
```

Trace the Working Code

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "a b temp\n" << "--------\n";

    int a = 5;      cout << a << endl;
    int b = 7;      cout << a << " " << b << endl;

    int temp = a;   cout << a << " " << b << "  " << temp << endl;
    a = b;          cout << a << " " << b << "  " << temp << endl;
    b = temp;       cout << a << " " << b << "  " << temp << endl;

    return 0;
}
```

```
a b temp
--------
5
5 7
5 7  5
7 7  5
7 5  5
```

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "a b c d\n" << "--------\n";

    int a, b, c;

    a = b = c = 3;  cout << a << " " << b << " " << c << endl;
    b = c = 5;      cout << a << " " << b << " " << c << endl;
    c = 7;          cout << a << " " << b << " " << c << endl;

    int d = b;      cout << a << " " << b << " " << c << " " << d << endl;

    return 0;
}
```

```
a b c d
--------
3 3 3
3 5 5
3 5 7
3 5 7 5
```

Circle the Variables' Scopes

```
#include <iostream>
using namespace std;

int main() {
    int a = 3, b = 3;
    cout << "a: " << a << "  b: " << b << endl;
    {
        int a = 5;
        b = a;
        cout << "a: " << a << "  b: " << b << endl;
    }
    cout << "a: " << a << "  b: " << b << endl;

    return 0;
}
```

```
a: 3  b: 3
a: 5  b: 5
a: 3  b: 5
```

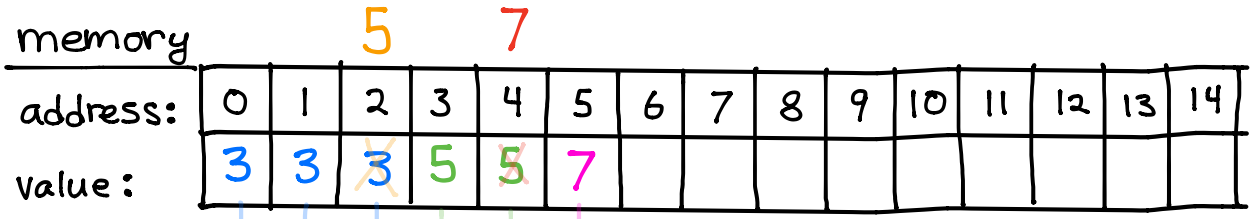
```cpp
#include <iostream>
using namespace std;

int main() {
    int a = 3, b = 3, c = 3;
    cout << "a: " << a << "  b: " << b << "  c: " << c << endl;
    {
        int a = 5, b = 5;
        c = b;
        cout << "a: " << a << "  b: " << b << "  c: " << c << endl;
        {
            int a = 7;
            b = a;
            cout << "a: " << a << "  b: " << b << "  c: " << c << endl;
        }
        cout << "a: " << a << "  b: " << b << "  c: " << c << endl;
    }
    cout << "a: " << a << "  b: " << b << "  c: " << c << endl;

    return 0;
}
```

```
a: 3  b: 3  c: 3
a: 5  b: 5  c: 5
a: 7  b: 7  c: 5
a: 5  b: 7  c: 5
a: 3  b: 3  c: 5
```

fill in da mems




```
#include <iostream>
using namespace std;

int main() {
    int a = 3, b = 3, c = 3;

    {
        int a = 5, b = 5;
        c = b;

        {
            int a = 7;
            b = a;
        }
    }

    return 0;
}
```

2 Data Types and Expressions

Fix the Broken Code

Trace the Working Code

Circle the Variables' Scopes

Fill in the Memory

3 If and If-Else

Fix the Broken Code

Trace the Working Code

Circle the Variables' Scopes

Fill in the Memory

4 Boolean Expressions

Fix the Broken Code

Trace the Working Code

Circle the Variables' Scopes

Fill in the Memory

5 Predefined Functions

Fix the Broken Code

Trace the Working Code

Circle the Variables' Scopes

Fill in the Memory

6 Loops

Fix the Broken Code

Trace the Working Code

Circle the Variables' Scopes

Fill in the Memory

7 Arrays

Fix the Broken Code

Trace the Working Code

Circle the Variables' Scopes

Fill in the Memory

8 Selection Sort

Fix the Broken Code

Trace the Working Code

Circle the Variables' Scopes

Fill in the Memory