Instructions

For a each topic (if applicable) there are 5 sections:

Short Answer

Answer each question as well as you can.

Circle the Variables' Scopes

Circle the scope of the given variables – i.e., circle the part of the code in which each 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").

Fix the Broken Code

Modify the code in the cleanest (and 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.

1 Variables and Assignment

Short Answer

Question

Describe undeclared variables, uninitialized variables, and initialized variables.

Answer

Question

Describe what we mean when we refer to a variable's scope.

Answer

Circle the Variables' Scopes

```cpp
#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;
}
```

Fill in the Memory

memory

address:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value:															

```cpp
#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;
}
```

Fix the Broken Code

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

int main() {
    // 5 mistakes

    int a = 5;
    int b = 7;

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

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

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

    return 0;
}
```

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

int main() {
    // 1 mistake

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

    int u;

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

    return 0;
}
```

Trace the Working Code

```
#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;
}
```

```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;
}
```

```
#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;
}
```

```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;
}
```

2 Data Types and Expressions

Short Answer

Question

What are the 5 main datatypes we studied during this course, and what is each typically used for?

Answer

Question

Give examples of what we mean by “order of operations’ – why does it matter? What is the order of operations for the following common operators:
`!=`, `==`, `=`, `<`, `>=`, `+`, `&&`, `||`, `*`, `()`, `>`, `/`, `<=`, `-`, `%`.

Answer

Fix the Broken Code

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

int main() {
    // 1 mistake

    int a = 3;
    cout << "One third is: " << (1 / a) << endl;

    double b = 5;
    cout << "One fifth is: " << (1 / b) << endl;

    return 0;
}
```

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

int main() {
    // 1 line has mistakes

    // formula (in math notation): C = (F-32)*(5/9)

    double fahrenheit = 100;  // getting close to summer!

    cout << "The temperature is:\n";
    cout << "    " << fahrenheit << " degrees Fahrenheit\n";
    cout << "    " << fahrenheit-32 * 5/9
         << " degrees Celcius\n";

    return 0;
}
```

Trace the Working Code

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

int main() {
    int a = 3, b = 5, c = 7;

    cout << ( 23%17 || ( -c && c != a && !!11 >= -a ) || 11/3 ) << endl;

    return 0;
}
```

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

int main() {
    int a = 3, b = 5, c = 7;

    cout << ( b == +a == b <= !c > !(13%17) <= +c - ( a == c ) ) << endl;

    return 0;
}
```

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

int main() {
    // 'a' == 97
    cout << ('a' + 5) << endl;
    cout << char('a' + 5) << endl;

    cout << 7 << endl;
    cout << (7/3) << endl;
    cout << double(7) << endl;

    cout << 3.14159265359 << endl;
    cout << int(3.14159265359) << endl;
    cout << char( 100 + int(3.14159265359) ) << endl;

    cout << (!53) << endl;
    cout << bool(5) << endl;

    cout << "this is a string" << endl;

    cout << (3 + 5 * 3 + 5 / 2) << endl;

    cout << (10 < 5 < 7) << endl;

    return 0;
}
```

3 If and If-Else

Fix the Broken Code

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

int main() {
    // 2 mistakes

    int a = 3;

    if (0 < a < 2)
        cout << "true" << endl
    else
        cout << "false" << endl;

    return 0;
}
```

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

int main() {
    // 1 mistake

    int a = 3;

    if (a = 5)
        cout << "true" << endl;
    else
        cout << "false" << endl;

    return 0;
}
```

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

int main() {
    // 1 type of mistake

    int a = 3;

    if (a = 5)
        cout << "true" << endl; a++;
    else
        cout << "false" << endl; a--;

    cout << "a = " << a << endl;

    return 0;
}
```

Trace the Working Code

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

int main() {
    srand(time(NULL));

    int a = rand()%10, b = rand()%10, c = rand()%10;

    cout << "a = " << a << ", b = " << b << ", c = " << c << endl;

    if (a <= b && a <= c)
        cout << "a is smallest" << endl;

    return 0;
}
```

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

int main() {
    srand(time(NULL));

    int a = rand()%10, b = rand()%10, c = rand()%10;
    int smallest = 0;

    cout << "a = " << a << ", b = " << b << ", c = " << c << endl;

    if (b < c)
        smallest = b;

    if (c <= b)
        smallest = c;

    if (a <= b && a <= c) {
        smallest = a;
        cout << "a is smallest" << endl;
    }

    return 0;
}
```

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

int main() {
    srand(time(NULL));

    int a = rand()%10, b = rand()%10, c = rand()%10;

    cout << "a = " << a << ", b = " << b << ", c = " << c << endl;

    if (a <= b && a <= c)
        cout << "a is smallest" << endl;
    else if (b <= a && b <=c)
        cout << "b is smallest" << endl;
    else
        cout << "c is smallest" << endl;

    return 0;
}
```

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

int main() {
    srand(time(NULL));

    int a = rand()%10, b = rand()%10, c = rand()%10;
    int smallest;

    cout << "a = " << a << ", b = " << b << ", c = " << c << endl;

    if (a <= b && a <= c) {
        smallest = a;
        cout << "a is smallest" << endl;
    } else if (b <= a && b <=c) {
        smallest = b;
        cout << "b is smallest" << endl;
    } else {
        smallest = c;
        cout << "c is smallest" << endl;
    }

    cout << "smallest = " << smallest << endl;

    return 0;
}
```

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

int main() {
    srand(time(NULL));

    int a = rand()%10, b = rand()%10, c = rand()%10;
    int smallest;

    cout << "a = " << a << ", b = " << b << ", c = " << c << endl;

    if (a <= b && a <= c) {
        smallest = a;
        cout << "a is smallest" << endl;
    } else if (b <= a && b <=c) {
        smallest = b;
        cout << "b is smallest" << endl;
    } else if (c <= a && c <= b) {
        smallest = c;
        cout << "c is smallest" << endl;
    }

    cout << "smallest = " << smallest << endl;

    return 0;
}
```

4 Functions

Short Answer

Question

What is a function? What is a predefined function? Why do we care?

Answer

Question

What library contains the `pow()` function? What are two other functions this library contains?

Answer

Fix the Broken Code

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

int main() {
    // 6 incorrect operators
    // 6 incorrect delimiters

    int a = 3, b = 5, c = 7;

    cout >> a >> '^' >> b >> ' (mod ' >> c >> ') = '
         >> int(pow(a,b)) % c >> endl;

    return 0;
}
```

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

int main() {
    // 4 mistakes

    int a = 3,  b = 5
        c = 7,  d == 11;
    double e = 13.3, f = 17.7;

    cout << "the sin of " << a << " is " << sin(a) << endl
         << "the cosine of " << b << " is " << sin(b) << endl
         << "the tangent of " << c << " is " << sin(c) << endl;

         << "the natural log of " << d << " is " << log(d) << endl
         << "the ceil of " << e << " is " << ceil(e) << << endl
         << "the floor of " << f << " is " << floor(f) << endl;

    return 0;
}
```

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


// This function prints "hello world!"
//
// Arguments: none
// Returns: nothing
void print_hello() {
    cout << "hello world!" << endl;
}


int main() {
    // 1 mistake

    cout << "now we are going to print \"hello world!\"\n";

    print_hello;

    return 0;
}
```

5 Loops

Short Answer

Question

What are the three types of loops in C++? Explain what each does, or what you might use it for.

Answer

Question

Is it ever okay to have an infinite loop? What usually causes unintentional infinite loops?

Answer

Circle the Variables' Scopes

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

int main() {
    int a = 3;

    for (int i = 7; i > a; i--) {

        for (int j = 11; j > i; j--) {

            cout << "*";

        }

        cout << endl;

    }

    return 0;
}
```

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

int main() {
    int a = 3;

    int i = 7;
    while (i > a) {

        int j = 11;
        while (j > i) {

            cout << "*";
            j--;

        }

        cout << endl;
        i--;

    }

    return 0;
}
```

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

int main() {
    // have to be a little careful with do-while loops:
    // it's not always possible to write one that means the
    // same thing as the corresponding for or while loop

    int a = 3;

    int i = 7;
    do {

        int j = 11;
        do {

            cout << "*";
            j--;

        } while (j > i);

        cout << endl;
        i--;

    } while (i > a);

    return 0;
}
```

Fix the Broken Code

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

int main() {
    // 2 mistakes

    cout << "i = ";
    for (int i=0; i<5; i+1) {
        cout << i << ", ";
    }
    cout << endl;

    cout << "i*2 = ";
    for (int i=0; i<5; i+1) {
        cout << i*2 << ", ";
    }
    cout << endl;

    return 0;
}
```

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

int main() {
    // 1 mistake

    int a = 0;
    while (a < 5) {
        cout << "a = " << a << endl;
    }

    return 0;
}
```

Trace the Working Code

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

int main() {
    int a = 3;

    for (int i = 7; i > a; i--) {

        for (int j = 11; j > i; j--) {

            cout << "*";

        }

        cout << endl;

    }

    return 0;
}
```

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

int main() {
    int a = 3;

    int i = 7;
    while (i > a) {

        int j = 11;
        while (j > i) {

            cout << "*";
            j--;

        }

        cout << endl;
        i--;

    }

    return 0;
}
```

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

int main() {
    // have to be a little careful with do-while loops:
    // it’s not always possible to write one that means the
    // same thing as the corresponding for or while loop

    int a = 3;

    int i = 7;
    do {

        int j = 11;
        do {

            cout << "*";
            j--;

        } while (j > i);

        cout << endl;
        i--;

    } while (i > a);

    return 0;
}
```

6 Arrays

Fill in the Memory

memory

address:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value:															

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

int main() {
    int a[5];

    int b[3] = {};

    int c[7] = {1,2,3};

    return 0;
}
```

memory

address:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value:															

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

int main() {
    int a[2][3] = {};

    int b[2][3] = {1, 2, 3, 4, 5, 6};

    return 0;
}
```

memory

address:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value:															

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

int main() {
    int a[2][3] = { {1, 2, 3}, {4, 5, 6} };

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

    return 0;
}
```

memory

address:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value:															

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

int main() {
    int a[2][3][2] = { { {1, 2}, {3, 4}, {5, 6} },
                       { {7, 8}, {9, 0}, {1, 2} } };

    return 0;
}
```

memory

address:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
value:															

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

int main() {
    int a[2][3][2] = { {1}, {2} };

    return 0;
}
```

Fix the Broken Code

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

int main() {
    // 2 mistakes

    a[7] = {1, 3, 5, 7, 9, 11, 13};

    cout << "a = { ";

    for (int i=0; i<=7; i++)
        cout << a[i] << ", ";

    cout << "};\n";

    return 0;
}
```

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

int main() {
    // 2 mistakes

    a[7] = {1, 3, 5, 7, 9, 11, 13};

    int b = a[7];

    cout << "b = " << b << endl;

    return 0;
}
```

Trace the Working Code

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

int main() {
    int a[7];

    for (int i=0; i<7; i++)
        a[i] = (i*7)%5;

    // print the array
    cout << "a = { ";
    for (int i=0; i<7; i++)
        cout << a[i] << ", ";
    cout << "};\n";

    return 0;
}
```

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

int main() {
    int a[7];

    for (int i=0; i<7; i++)
        cin >> a[i];

    // print the array (to see what the user entered)
    cout << "a = { ";
    for (int i=0; i<7; i++)
        cout << a[i] << ", ";
    cout << "};\n";

    return 0;
}
```

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

int main() {
    int a[7] = { 6, 7, 3, 5, 2, 1, 4 };

    int sum = 0;
    for (int i=0; i<7; i++)
        sum += a[i];

    cout << "the average of these elements is " << sum/7.0 << endl;

    return 0;
}
```

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

int main() {
    int a[7] = { 6, 7, 3, 5, 2, 1, 4 };

    int min = a[0];
    int max = a[0];

    for (int i=1; i<7; i++) {

        if (a[i] < min)
            min = a[i];

        if (a[i] > max)
            max = a[i];
    }

    cout << "the smallest of these elements is " << min << endl;
    cout << "the largest of these elements is " << max << endl;

    return 0;
}
```