Supplemental Instruction Handout

Break up each snippet of code by highlighting the important parts of the code. If a part is referenced again, highlight it again with the same color. Sample provided below.

Breaking Down the Program

Sample Code:

```
//Example: scientific and fixed numbers in a program

#include <iostream>
using namespace std;

int main()
{
      double hours = 35.45;       //notice that variables are
      double rate = 15.00;        //highlighted in green, literals are
      double tolerance = 0.01000; //in yellow, and stream manipulators
                                  //are in purple
      cout << "hours = " << hours << ", rate = " << rate
           << ", pay = " << hours * rate
           << ", tolerance = " << tolerance << endl << endl;
      cout << scientific;
      cout << "Scientific notation: " << endl;
      cout << "hours = " << hours << ", rate = " << rate
           << ", pay = " << hours * rate
           << ", tolerance = " << tolerance << endl << endl;
      cout << fixed;
      cout << "Fixed decimal notation: " << endl;
      cout << "hours = " << hours << ", rate = " << rate
           << ", pay = " << hours * rate
           << ", tolerance = " << tolerance << endl << endl;
      return 0;
}
```

What do you predict this program will output? (Explain in words)

It's probably going to display the given variables in scientific notation then in fixed point notation.

Supplemental Instruction Handout

Break up each snippet of code by highlighting the important parts of the code. If a part is referenced again, highlight it again with the same color. Sample provided below.

Snippet #1:

```
// This program illustrates that a value-returning function
// returns only one value, even if the return statement
// contains more than one expression. This is a legal, but not
// a recommended code.
#include <iostream>
using namespace std;
int funcRet1();
int funcRet2(int z);

int main()
{
      int num = 4;
      cout << "Line 1: The value returned by funcRet1: "
      << funcRet1() << endl;            // Line 1
      cout << "Line 2: The value returned by funcRet2: "
      << funcRet2(num) << endl;         // Line 2
      return 0;
}

int funcRet1()
{
      int x = 45;
      return 23, x;                 //only the value of x is returned
}

int funcRet2(int z)
{
      int a = 2;
      int b = 3;
      return 2 * a + b, z + b;    //only the value of z + b is returned
}
```

What do you predict this program will output? Why is this a bad program?(Explain in words)

Line 1: The value returned by funcRet1: 45
Line 2: The value returned by funcRet2: 7

This is bad because it wants the functions to return two values; functions can only return one value.

Supplemental Instruction Handout

Break up each snippet of code by highlighting the important parts of the code. If a part is referenced again, highlight it again with the same color. Sample provided below.

Snippet #2:

```
//Program: Largest of three numbers
#include <iostream>
using namespace std;

double larger(double x, double y);
double compareThree(double x, double y, double z);

int main()
{
      double one, two;                        //Line 1

      cout << "Line 2: The larger of 5 and 10 is "
           << larger(5, 10) << endl;          //Line 2
      cout << "Line 3: Enter two numbers: ";//Line 3
      cin >> one >> two;                      //Line 4
      cout << endl;                           //Line 5
      cout << "Line 6: The larger of " << one
           << " and " << two << " is "
           << larger(one, two) << endl;       //Line 6
      cout << "Line 7: The largest of 43.48, 34.00, "
           << "and 12.65 is "
           << compareThree(43.48, 34.00, 12.65)
           << endl;                           //Line 7
      return 0;
}
double larger(double x, double y)
{
      double max;
      if (x >= y)
            max = x;
      else
            max = y;
      return max;
}
double compareThree (double x, double y, double z)
{
      return larger(x, larger(y, z));
}
```

Supplemental Instruction Handout

Break up each snippet of code by highlighting the important parts of the code. If a part is referenced again, highlight it again with the same color. Sample provided below.

What do you predict the previous program will output? (Explain in words)