Floating-point Output Formats Exercises

Floating-point output

- Write a program that executes the code below

```
double pi {3.1415926535};
cout << pi << endl;

double c {299'792'458};
cout << c << endl;
```

- Explain the results

Scientific Notation

- The following numbers are in scientific notation
- What are their values in "everyday" notation?

  2.99792E+008

  1.00000E-006

  3.14159E+000

Scientific Manipulator

- What effect does the "scientific" manipulator have?
- What effect does the "uppercase" manipulator have?
- What output does the following code give?

```
double pi {3.1415926535};
cout << scientific << pi << endl;
cout << scientific << uppercase << pi << endl;
```

Fixed-point Output

- What is meant by "fixed-point" output?
- What output does the following code give?

```
double c {299'792'458};
cout << fixed << c << endl;
double e {1.602e-19};
cout << fixed << e << endl;
```

Restoring Floating-point Defaults

- Why is it important to restore the defaults for floating-point output?
- How do we restore these defaults?

Floating-point Precision

- What is meant by floating-point precision?
- How can we find the current value of the floating-point precision?
- How can we set the floating-point precision?