The *output manipulators* `dec`, `hex`, and `oct` are used for converting different bases, as the next example illustrates.

EXAMPLE G.1 Using Output Manipulators

This shows how both the value and the address of a variable can be printed:

```
int main()
{ int n = 1492;  //  base 10
  cout << "Base  8: n = " << oct << n << endl;
  cout << "Base 10: n = " << n << endl;
  cout << "Base 16: n = " << hex << n << endl;
}
```

```
Base  8: n = 2724
Base 10: n = 1492
Base 16: n = 5d4
```

Here the manipulator `oct` is used to convert the next output to octal form. Note that the output reverts back to decimal until the `hex` manipulator is used.

The next example shows how to input integers in octal and hexadecimal. Octal numerals are denoted with a `0` prefix, and hexadecimal numerals are denoted with a `0x` prefix.

EXAMPLE G.2 Using Input Manipulators

This shows how both the value and the address of a variable can be printed:

```
int main()
{ int n;
  cout << "Enter an octal numeral (use 0 prefix): ";
  cin >> oct >> n;
  cout << "Base  8: n = " << oct << n << endl;
  cout << "Base 10: n = " << dec << n << endl;
  cout << "Base 16: n = " << hex << n << endl;
  cout << "Enter a decimal numeral: ";
  cin >> dec >> n;
  cout << "Base  8: n = " << oct << n << endl;
  cout << "Base 10: n = " << dec << n << endl;
  cout << "Base 16: n = " << hex << n << endl;
  cout << "Enter a hexadecimal numeral (use 0x prefix): ";
  cin >> hex >> n;
  cout << "Base  8: n = " << oct << n << endl;
  cout << "Base 10: n = " << dec << n << endl;
  cout << "Base 16: n = " << hex << n << endl;
}
```

```
Enter an octal numeral (use 0 prefix): 0777
Base  8: n = 777
Base 10: n = 511
Base 16: n = 1ff
Enter a decimal numeral: 511
Base  8: n = 777
Base 10: n = 511
```