<iomanip>

`setiosflags (ios_base::fmtflags mask)`	Sets the format flags specified by parameter mask. Basically, the input is a bunch of format flags binary or'd together. `std::cout << std::setiosflags (std::ios::showbase | std::ios::uppercase);`
`resetiosflags(ios_base::fmtflags mask)`	Unsets the format flags specified by parameter mask.
`setprecision (int n)`	Sets the decimal precision to be used to format floating-point values on output operations.
`setw(int n)`	Behaves as if member width were called with n as argument on the stream on which it is inserted/extracted as a manipulator.
`setfill (char_type c)`	Sets c as the stream's fill character.
`get_money (moneyT& mon, bool intl = false)`	Extracts characters from the input stream it is applied to, and interprets them as a monetary expression, which is stored as the value of mon.
`put_money (const moneyT& mon, bool intl = false)`	Inserts the representation of mon as a monetary value into the output stream it is applied to.

<iomanip> (cont)

`get_time (struct tm tmb, const charT fmt)`	Extracts characters from the input stream it is applied to, and interprets them as time and date information as specified in argument fmt. The obtained data is stored at the struct tm object pointed by tmb.
`put_time (const struct tm tmb, const charT fmt)`	Inserts the representation of the time and date information pointed by tmb, formatting it as specified by argument fmt.

The return value for each of the above is unspecified, and they should only be used as stream manipulators.

Money Manipulators

```
#include <iostream> // std::cin,
std::cout
#include <iomanip> //
std::get_money
int main ()
{
  long double price;
  std::cout << "Please, enter the
price: ";
  std::cin >>
std::get_money(price);
  if (std::cin.fail()) std::cout <<
"Error reading price\n";
  else std::cout << "The price
entered is: " << price << '\n';
  return 0;
}
```

The