

Quizzes



Name:

Name three member functions from **MP3** class used in lab 1. Describe what each function does including its parameters (if anything) and what it returns (if anything).

Listed below is a portion of the `Rational` class that we have been developing this past week. Implement the `extract` member function which should accept an input stream and read a rational number from it. You may assume that the rational number is of the form:

```
1/8
2 1 /8
   1 / 8
4 or
   1 / 8
```

Place the prototype in the appropriate place and implement the function on the back of this sheet of paper.

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

4 class Rational {
  public:
6   Rational(long int numer=0, long int denom=1);
   Rational(const Rational& org);
8   ~Rational();
   Rational& operator=(const Rational& rhs);
10
12
  private:
14   reduce();

16   long int num;
18   unsigned long int den;
};
```

Listed below is a portion of the Rational class that we have been developing this quarter.

```

#include <iostream>
2 using namespace std;

4 class Rational {
public:
6   Rational(long int numer=0, long int denom=1);
   Rational(const Rational& org);
8   ~Rational();
   Rational& operator=(const Rational& rhs);
10  Rational operator+(const Rational& rhs) const;
   Rational operator-(const Rational& rhs) const;
12  Rational operator/(const Rational& rhs) const;
   Rational operator*(const Rational& rhs) const;
14  void display() const;
   bool extract(istream& is);
16  Rational& operator+=(const Rational& rhs);
   Rational& operator-=(const Rational& rhs);
18  Rational& operator/=(const Rational& rhs);
   Rational& operator*=(const Rational& rhs);
20  bool operator!=(const Rational& rhs) const;
   bool operator==(const Rational& rhs) const;
22  bool operator<(const Rational& rhs) const;
private:
24  reduce();
   long int num;
26  unsigned long int den;
};

28
30 Rational operator+(long int lhs, const Rational& rhs);
   Rational operator-(long int lhs, const Rational& rhs);
32 Rational operator/(long int lhs, const Rational& rhs);
   Rational operator*(long int lhs, const Rational& rhs);
   ostream& operator<<(ostream& os, const Rational& rhs);
34 istream& operator>>(istream& is, const Rational& rhs);

```

Indicate (with line numbers) which of the above functions are called directly in the following code. You should write the line number for each function call. Answers for the first two lines of code are given in comments to the right of each line.

```

Rational a(4,7); // 6
2 a = a - a - a - a; // 11, 11, 11, 9
   Rational b = a; // -----
4 cout << "Enter a rational number:\n"; // -----
   cin >> b; // -----
6 const Rational c(a+b); // -----
   cout << a << "+\n" << b << "=\n" << c << endl; // -----
8 a = 1; // -----
   while(a!=0) { // -----
10  cout << "Enter any rational number other than zero (or zero to quit):\n";
     Rational num; // -----
12  cin >> num; // -----
     cout << a << "*\n" << num << "=\n"; // -----
14  a *= b; // -----
     cout << a; // -----
16 }

```

Implement the assignment operator for the following simple container class that holds `double`s. If the vector has zero elements, you should not dynamically allocate any memory.

```
class VectorD {
2   VectorD ();
   VectorD(const VectorD& rhs);
4   ~VectorD ();
   VectorD& operator=(const VectorD& rhs);
6   unsigned int size() const;
   const char& operator[](unsigned int i) const;
8   char& operator[](unsigned int i);
   void insertion(ostream& os) const;
10 private:
   unsigned int bigness;
12   double* numbers;
   };
14 ostream& operator<<(ostream& os, const VectorD& str);
```

Quizzes



Name:

Explain the difference between function **overriding** and function **overloading**.

Indicate what will displayed when the following code is executed. Please show your reasoning.

```
#include <iostream>
2 using std::cout;

4 void function(int index);

6 int main()
  {
8   int i = 0;
  cout << i++ << '\n';
10  function(i);
  try {
12   cout << ++i << '\n';
  function(i);
14   cout << i++ << '\n';
  function(i);
16  }
  catch(double except) {
18   function(except-1);
  }
20  catch(...) {
  cout << --i << '\n';
22  }
  cout << i++ << '\n';
24  function(i);
  cout << i++ << '\n';
26
  return 0;
28 }

30 void function(int index)
  {
32   if(index>2) {
  throw index;
34  }
  }
```