

Consider the following code segment:

```
int i = 4;  
2 int* ptr1;  
int* ptr2 = 0;  
4 int* ptr3 = &i;  
int j = i++;  
6 ptr1 = ptr3;  
cout << *ptr1;
```

Using a drawing similar to the ones done in lecture, indicate what has taken place in memory after the first 5 lines (including line 5) of code have been executed. Also indicate the value that will be displayed by the statement on line 7.



Describe what the \* operator and & operator do. How are they different?

The function prototypes from the header file for the Disc class used in the first lab assignment are shown below.

```
class Disc
2 {
  public:
4   Disc();
   Disc(const Disc& rhs);
6   ~Disc();
   Disc& operator=(const Disc& rhs);
8   string getTitle() const;
   string getArtist() const;
10  string operator[](int i) const;
   unsigned int getLength() const;
12  unsigned int getNumTracks() const;
   bool readDisc(istream& is);
14  bool operator==(const Disc& rhs) const;
   bool operator<(const Disc& rhs) const;
16  void streamInsert(ostream& os) const;
  private:
18  void fixString(string& line) const;
   unsigned int length;
20  string title;
   string artist;
22  vector<string> tracks;
};
24  ostream& operator<<(ostream& os, const Disc& album);
```

Implement the extraction operator so that it is possible to do the following:

```
Disc cd;
2 cin >> cd;
```

Consider the following definition for the `Rational` class. Using the same requirements about the data members that we made in class, give the implementation for the member functions on lines 8 and 12. You may assume that the rest of the functions listed have already been implemented.

```
class Rational {
2 public:
    Rational();
4    Rational(const Rational& rhs);
    Rational(int numer, int denom=1);
6    ~Rational();
    Rational add(const Rational& rhs) const;
8    Rational subtract(const Rational& rhs) const;
    Rational multiply(const Rational& rhs) const;
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);
14 private:
    void reduce();
16   int num;
    unsigned int den;
18 };
```



There are four functions that will automatically be generated by the compiler if they are not listed in the class definition. Name these four functions.

Suppose that the following class has been defined.

```

2  enum Months={January=1, February, March, April, May, June, July,
    August, September, October, November, December};
3  class Date {
4  public:
    Date();
6   Date(unsigned int mn, unsigned int day, int yr);
    Date(const Date & rhs);
8   Date& operator=(const Date & rhs);
    ~Date();
10  unsigned int getMonth() const;
    unsigned int getDay() const;
12  int getYear() const;
    bool setDate(unsigned int mn, unsigned int yd, int yr);
14  Date& operator++();
    Date& operator--();
16  void display(ostream & os) const;
    bool read(istream & is);
18  bool isLeapYear() const;
    string dayOfWeek() const;
20 private:
    Months month;
22  unsigned int day;
    int year;
24  vector<unsigned int> numDays;
    bool valid() const;
26 };

```

Indicate which member functions are called on each line of the following code segment:

```

2   Date today(3,28,2001);
3
4
5   Date tomorrow=today;
6
7
8   ++tomorrow;
9
10
11
12  cout << "Today is a_" << today.dayOfWeek() << endl;
13
14
15
16  Date nextMonth(today.getDay(), today.getMonth()+1, today.getYear());
17
18
19
20  cout << "In a month it will be a_" << nextMonth.dayOfWeek() << endl;

```

Quizzes



Name:

---

Explain the difference between inheritance and composition.

Consider the following definition for the `String` class we developed in lecture. Implement the concatenate operator prototyped on line 12. Note: the names of the data members have changed.

```
class String {
2 public:
    String();
4    String(const String& rhs);
    String(char [] str);
6    ~String();
    String& operator=(String rhs);
8    unsigned int size() const;
    // Concatenate operator: adds the String passed, str, to the
10   // current String
    // e.g., String name("Jon"); String last("Dough");
12   // name += last; // "JonDough"
    String& operator+=(String rhs);
14
16 private:
    unsigned int numEl;
    char* letters;
18 };
```



Quizzes



Name:

---

Write a templated function that accepts three items and returns the median of the three.



Explain what overriding a member function means.

Quizzes



Name:

---

What is an abstract base class? What purpose do abstract base classes serve?

Quizzes



Name:

---

Write a templated function that accepts three items and returns the median of the three.



Explain the effect of declaring a function `virtual`. Also, is it necessary to override a `virtual` function in any derived classes?