Instructions: Opscan forms will be passed out in class. Write your name and code your ID number on the opscan form.
Turn in your completed opscan at class on Thursday Sept 19. No late opscans will be accepted.

For questions 1 and 2, consider the class declaration:

```
class CreditCard {
public:
    CreditCard(double Amount) { Balance = Amount; }
    void   Payment( /* in */ double Amount ) { Balance -= Amount; }
    void   Charge( /* in */ double Amount ) {Balance += Amount; }
    double CardBalance() const { return Balance; }
private:
    double Balance;
};
```

1.   Which member function provides an observer operation?

1) Payment	3) CardBalance	5) 1, 2, and 3
2) Charge	4) 1 and 2 only	6) None of these

2.   Which member function provides a mutator operation?

1) Payment	3) CardBalance	5) 1, 2, and 3
2) Charge	4) 1 and 2 only	6) None of these

3.   Consider the following statements, assuming the class declaration for CreditCard is in scope:

```
CreditCard WesternUnion(1000.0);
CreditCard BankAmerica(0.0);
WesternUnion.Payment(400.0);        // line 1
cout << WesternUnion.Balance;       //      2
BankAmerica = WesternUnion;         //      3
CreditCard.Charge(120.0);           //      4
```

Which statements would cause compilation errors?

1) 1 only	5) 1 and 3 only	9) 2, 3 and 4 only
2) 2 only	6) 1 and 4 only	10) None of these
3) 3 only	7) 2 and 3 only	
4) 4 only	8) 2 and 4 only	

4.   Which of the following statements about C++ classes is *false*?

1) Classes can have private member functions.	4) Classes can have public data members.
2) Classes can have public, private and protected members.	5) Aggregate assignment is permitted for classes.
3) By default, members of classes are public.	6) None of these, (all are true).

5.   Which of the following C++ built-in operations are defined for class objects?

1) =	4) 1 and 2 only	7) All of them
2) ==	5) 1 and 3 only	8) None of these
3) <<	6) 2 and 3 only	

6.   A class `A` has a public member function `G` that takes one `int` parameter for input, returns a `bool` value, and does not
      modify any of the class data members. Which of the following would be the best correct function prototype for G in the
      class declaration?

      1) `bool G(int x) const;`   4) `const bool G(int& x);`
      2) `const bool G(int x);`   5) `bool G(int& x) const;`
      3) `bool G(int& x) const;`   6) None of these

7.   Suppose that the declaration of the class `A` includes the following function prototype.

      ```
      bool LessThan( /* in */ const A& RHS );
      ```

      Which of the following tests in the client code correctly compares two `A` objects named `Alpha` and `Beta`?

      1) `if (Alpha < Beta)`   4) `if (Alpha.LessThan.Beta)`
      2) `if (Alpha.LessThan(Beta))`   5) `if (LessThan(Alpha).Beta)`
      3) `if (LessThan(Alpha, Beta))`   6) None of these

8.   If the designer of a C++ class wishes to allow clients to inspect but not modify private data, what is the best approach?

      1) Provide an observer function as a class member.
      2) Provide a mutator function as a class member.
      3) Declare the data to be public, not private.
      4) Provide an additional class constructor.
      5) Do nothing because it is not acceptable to let clients inspect private data.
      6) None of these

Consider the following class:

```
class B {
private:
   int S;
public:
   B(int initS) { S = initS; }
};
```

9.   Assuming everything necessary is in scope, consider the declaration: `B Foo;`

      What is the value of `Foo.S`?

      1) 0   3) The declaration isn't allowed.
      2) Unknown   4) None of these

10.   Assuming everything necessary is in scope, consider the code fragment:

      ```
      B Foo(17), Bar(32);
      Bar = Foo;
      ```

      What is the value of `Bar.S`?

      1) 17   3) Unknown   5) None of these
      2) 32   4) The assignment isn't allowed.

For questions 11 through 16, consider the class declaration:

```
class Farey {
private:
    int Top, Bottom;
public:
    Farey();
    Farey(int T, int B);
    Farey operator+(const Farey& RHS) const;
    Farey operator-(const Farey& RHS) const;
    bool  operator==(const Farey& RHS) const;
    void  Display(ostream& Out) const;
};

Farey::Farey() {
    Top = Bottom = 0;
}

Farey::Farey(int T, int B) {
    Top    = T;
    Bottom = B;
}

Farey Farey::operator+(const Farey& RHS) const {

    return Farey(Top + RHS.Top, Bottom + RHS.Bottom);
}

Farey Farey::operator-(const Farey& RHS) const {

    return Farey(Top - RHS.Top, Bottom - RHS.Bottom);
}

bool Farey::operator==(const Farey& RHS) const {

    return ( (Top == RHS.Top) && (Bottom == RHS.Bottom) );
}

void Farey::Display(ostream& Out) const {

    Out << Top << '/' << Bottom;
}
```

Again, assuming everything necessary is in scope, consider the following code fragment:

```
Farey A(3, 5), B(1, 4), C(2, 4), D(0, 5), E;
E = A + B;                        // line 1
A.Display(cout);                  //     2
E = A + B - C;                    //     3
E = 2*A;                          //     4
```

11.   After the execution of line 1, what are the values of E.Top and E.Bottom, respectively?

1) 0 and 0	3) 1 and 4	5) Unknown
2) 3 and 5	4) 4 and 9	6) None of these

12.   What is written to the stream `cout` when line 2 is executed?

   1) `"3/5"`   2) Nothing   3) None of these

13.   After the execution of line 3, what are the values of `E.Top` and `E.Bottom`, respectively?

   1) 0 and 0   3) 2 and 5   5) Unknown
   2) 4 and 9   4) The statement isn't allowed.   6) None of these

14.   After the execution of line 4, what are the values of `E.Top` and `E.Bottom`, respectively?

   1) 6 and 10   3) 3 and 10   5) Unknown
   2) 6 and 5   4) The statement isn't allowed.   6) None of these

Consider the following code fragment:

```
Farey X(1, 2), Y(2, 4);

if ( X == Y )                       // line 5
   cout << "X == Y" << endl;
else
   cout << "X != Y" << endl;

if ( X + X == Y )                   // line 6
   cout << "X + X == Y" << endl;
else
   cout << "X + X != Y" << endl;
```

15.   When the `if` statement beginning in line 5 is executed, what is written to `cout`?

   1) `"X == Y"`   2) `"X != Y"`

16.   When the `if` statement beginning in line 6 is executed, what is written to `cout`?

   1) `"X + X == Y"`   2) `"X + X != Y"`

For questions 17 through 20, consider the following class:

```
class Quadratic {
private:
   double Coefficient[3];
public:
   Quadratic(double a = 0.0, double b = 0.0, double c = 0.0);
   double Evaluate(double x) const;
};

Quadratic::Quadratic(double a, double b, double c) {

   Coefficient[0] = a;
   Coefficient[1] = b;
   Coefficient[2] = c;
```

```
   }

   double Quadratic::Evaluate(double x) const {

       return ( Coefficient[0]*x*x + Coefficient[1]*x + Coefficient[2] );
   }
```

17.   Given the declaration: `Quadratic F(1, 2, 3);`

What value is output by the statement: `cout << F.Evaluate(2);`

1) `6`   2) `11`   3) `17`   4)   None of these

18.   Given the declaration: `Quadratic G(1);`

What value is output by the statement: `cout << G.Evaluate(2);`

1) `1`   2) `4`   3)   Not allowed.   4)   None of these

A designer wants to add an addition operation to the class `Quadratic`. Consider the partial implementation:

```
   _____  Quadratic::operator+(const Quadratic& RHS) const {  // line 1

       double a = Coefficient[0] + RHS.Coefficient[0];
       double b = Coefficient[1] + RHS.Coefficient[1];
       double c = Coefficient[2] + RHS.Coefficient[2];

       return _____;                                           // line 2
   }
```

19.   How should the blank in line 1 be filled?

1) `void`   4)   It should be left blank.
2) `Sum`   5)   None of these
3) `Quadratic`

20.   How should the blank in line 2 be filled?

1) `Quadratic(a, b, c)`   4)   It should be left blank.
2) `Sum`   5)   None of these
3) `Quadratic(c, b, a)`