Physics 5 – Final Exam – Fall '08
Name_____

1. True/False: (4 points each)
   a. The **int** type will "wrap-around" when its point of overflow occurs.
   b. `char *msg = "Hello";` is equivalent to `char *msg = {'H','e','l','l','o','\0'};`

2. What would be printed by the following statements?

```
#include <iostream>

int f(int &i)
{
   i = 10;
   return(5 * i);
}

int main()
{
    int n = 5;
    f(n);
    cout << n << "\n";
    return 0;
}
```

3. What would be printed by the following code fragment`?

```
int y[3][3] = {1, 2, 3,
               4, 5, 6,
               7, 8, 9};
cout << y[1][2];
```

4. What would be printed by the following program?

```
#include <iostream>

int sub1(int &n)
{
  n--;
  return n;
}

int main()
{
  int m = 10;
  for(int j = 0; j < 10; j++)
    m -= sub1(j);
    cout << m << "\n";
  return 0;
}
```

5. What would be printed by the following statements?

```
double *pt;
double a[3] = {1.2, 2.3, 3.4};
pt = &a[1];
pt += 1;
cout << *pt << endl;
```

6. What would be printed by the following statements? Why do you think so?

```
int k;
double j;
k = 2;
j = 2.0;
if(k == j){
    cout << "Okay.";
}
else
    cout << "Not okay.";
```

7. Given a program as follows, what would be printed?

```
#include <iostream.h>

int myfunc(double);
int myfunc(float);

int main()
{
    cout << myfunc(3.51) << "\n";
    return 0;
}

int myfunc(double n)
{
    return n * 2.0;
}

int myfunc(float n)
{
    return n * 3.0;
}
```

8. Which of the following statements is equivalent to

   `pt->x_center = 10.0;` ?

   (a) `*pt.x_center = 10.0;`
   (b) `(*pt).x_center = 10.0;`
   (c) `(*pt.)x_center = 10.0;`
   (d) `(pt->)x_center = 10.0;`

9. Assuming the assignment operator has been appropriately overloaded, what would be printed by the statements following?

```
class Complex {
public:
     double re, im;
};

Complex x, y;

x.re = 4.0;
x.im = 5.0;

y = x;
x.re = 5.0;
cout << y.re << endl;
```

10. What would be printed by the following program?

```
#include <iostream>

int sum(int pt[], int n)
{
   int temp = 0;
   for (int i = 0; i < n; ++i) {
      temp += pt[i];
   }
   return temp;
}

int main()
{
   int total;
   int pt[5];
   for (int i = 0; i < 5; i++)
      pt[i] = i;
   total = sum(pt, 3);
   cout << total << " " << i <<
endl;
   return 0;
}
```

11. Which of the following statements contains a reference declarator?
(a) `int *i;`
(b) `swap(&x, &y);`
(c) `int x, *pt; pt = &x;`
(d) `int x, &pt = x;`
(e) None of the above.

12. What would be printed by the following statements? Why?

```
double x = !0 * (1 / 3 * 3.0);
cout << x << endl;
```

13. Explain how to dynamically (a) allocate and (b) deallocate a **`float`** array `X[5]` by using **`new`** and **`delete`** operators.

14.
```
int foo(int x)
{
   if(x == 0 || x == 1 || x == 2)
      return x;
   else if( x <= 5)
      return x*x - foo(x - 1);
   else
      return x/2 + foo(x - 1);
}
```

What are the return values for the following function calls? (Hint: trace each function call)

`foo(1)` _____

`foo(2)` _____

`foo(3)` _____

`foo(4)` _____

`foo(5)` _____

`foo(6)` _____

`foo(7)` _____

`foo(8)` _____

`foo(9)` _____

15. Please complete the following program (by filling in lines 2, 6, 17, 24, 27, 28, 38) so that it will read data from the file `test.dat` and print out the data.

```
1   #include <iostream.h>
2   _____________________
3
4   #define MAX 10
5
6   _________________________________
7   public:
8      char name[25];
9      int Ex1, Ex2;
11  };
11
12  int main()
13  {
14     Student st[MAX];
15     int count = 0;
16
17     _________________
18     if (!in)
19     {
20        cout << "Cannot open
                    test.dat.\n";
21        return 1;
22     }
23
24     _________________
25     while(!in.eof())
26     {
27        ___________________
28        ___________________
29        cout << st[count].name <<
          " " << st[count].Ex1 <<
30        " " << st[count].Ex2 <<
          "\n";
31        count++;
32        if (count >= 10)
33        {
34           cout << "Exceed MAX: "
                  << MAX << endl;
35           in.close();
36           return(-1);
37        }
38        ___________________
39     }
40
41     in.close();
42     return 0;
43  }
```

The input file test.dat contains the following data:

```
Eric 77 87
Scott 90 94
Mary 100 100
```

16. What is the output of the following code?

```
class Nmbr
{
private:
   int m_num;
public:
   Nmbr(int n);
   Nmbr();
   int GetNmbr();
   void SetNmbr(int n);
   Nmbr operator =(Nmbr n2);
   Nmbr operator *(Nmbr n2);
};

Nmbr::Nmbr()
{
   m_num = 0;
}
Nmbr::Nmbr(int n)
{
   m_num = n;
}
int Nmbr::GetNmbr()
{
   return m_num;
}
void Nmbr::SetNmbr(int n)
{
    m_num = n;
}
Nmbr Nmbr::operator *(Nmbr n2)
{
   return Nmbr(m_num * n2.m_num);
}
Nmbr Nmbr::operator =(Nmbr n2)
{
   m_num = n2.m_num * 2;
   return Nmbr(m_num);
}
int main()
{
   Nmbr n1(5), n2(3), n3;
   n3 = n1 * n2;
   cout << n3.GetNmbr() << endl;
   return 0;
}
```

Output:

Physics 5 – Final Exam Solutions – fall 08

1. True/False: (4 points each)

a. The **`int`** type will "wrap-around" when its point of overflow occurs.
SOLN:
True: consider the simple program to print out number starting with 2^{31}-2 and prints out subsequent integers:

```
int main() {
     int I = 2147483645;
     for(int k = 0; k < 10; k++)
       cout << I + k << endl;
}
```

The output shows the wrap-around:

```
2147483646
2147483647
-2147483648
-2147483647
-2147483646
```

b. **`char`** `*msg = "Hello";` is equivalent to **`char`** `*msg = {'H','e','l','l','o','\0'};`

SOLN: false: *msg is a pointer
When you declare a pointer variable, you get a block of memory that's large enough to hold the address in memory of something of a specific type.

In the first, msg is a variable that holds the address of a char, namely the character 'h' in the string constant "hello". String constants, unlike other constants, are stored in a part of memory called the string table. A string constant has a value--the address in memory of its first character.

Assuming for the moment that the characters in "hello" are stored at the address starting at 0x100, and that each character is stored in a single byte, the characters of this string will be at locations 0x100 through 0x105, which includes the ASCII NULL (i.e., 0) at the end.

On the other hand, this expression --- `{'H','e','l','l','o','\0'};` is treated differently: Somewhere along the line when C was defined, someone decided that arrays of char should be treated a little differently from all other array types. The same behavior was kept in C++. This special treatment of char arrays was probably for the sake of convenience at the expense of consistency with other array types, inasmuch as char arrays are probably the most common array type (this is the opinion of Kernighan and Ritchie).

2. What would be printed by the following statements?

```
#include <iostream>

int f(int &i)
{
   i = 10;
   return(5 * i);
}

int main()
{
    int n = 5;
    f(n);
    cout << n << "\n";
    return 0;
}
```

SOLN:
This is a bit tricky, huh? Follow the `n` and you'll see that, while `f` returns 50, it changes `n` to 10, so this will be the value output by the program.

3. What would be printed by the following code fragment`?

```
int y[3][3] = {1, 2, 3,
               4, 5, 6,
               7, 8, 9};
cout << y[1][2];
```

SOLN: This will be, just as it appears, the second row third column element: 6.

4. What would be printed by the following program?

```
#include <iostream>

int sub1(int &n)
{
  n--;
  return n;
}

int main()
{
  int m = 10;
  for(int j = 0; j < 10; j++)
    m -= sub1(j);
  cout << m << "\n";
  return 0;
}
```

SOLN: There is no output since this has an infinite loop. What happens is that, while the for loop is busy incrementing j, the function sub1() decreases it so it never gets anywhere: it's value is decreased once to -1 where it stays forever and ever and ever…and ever…

5. What would be printed by the following statements?

```
double *pt;
double a[3] = {1.2, 2.3, 3.4};
pt = &a[1];
pt += 1;
cout << *pt << endl;
```

SOLN: pt is initialized to the address of the second element of the array and then the address is incremented to the third element. The code then outputs the third element, which is 3.4.

6. What would be printed by the following statements? Why do you think so?

```
int k;
double j;
k = 2;
j = 2.0;
if(k == j){
    cout << "Okay.";
}
else
    cout << "Not okay.";
```

SOLN: "Okay" is the output. In this context there is a built in conversion of the int to a double so the different types can be compared and, if they're equal to within a truncation error, the comparison returns a true.

7. Given a program as follows, what would be printed?

```
#include <iostream.h>

int myfunc(double);
int myfunc(float);

int main() {
    cout << myfunc(3.51) << "\n";
    return 0;
}

int myfunc(double n) {
    return n * 2.0;
}

int myfunc(float n) {
    return n * 3.0;
}
```

SOLN: This is an example of function overloading. Since 3.51 could be either a float or a double, it goes for the first definition that fits: in this case that's the double type input, which returns the integer truncated value of 7.02, that is, it returns and prints 7 to the console.

8. Which of the following statements is equivalent to

```
pt->x_center = 10.0;
```

?

(a) `*pt.x_center = 10.0;`
(b) `(*pt).x_center = 10.0;`
(c) `(*pt.)x_center = 10.0;`
(d) `(pt->)x_center = 10.0;`

SOLN: (b) The arrow operator (->) is a dereference operator that is used exclusively with pointers to objects with members. This operator serves to access a member of an object to which we have a reference. Due to the order of operations, your need to dereference the pointer before you can access the members, leading to a sort of awkward looking `(*pt).x_center = 10.0;`

9. Assuming the assignment operator has been appropriately overloaded, what would be printed by the statements following?

```
class Complex {
public:
     double re, im;
};
Complex x, y;
x.re = 4.0;
x.im = 5.0;
y = x;
x.re = 5.0;
cout << y.re << endl;
```

SOLN: 4.0

10. What would be printed by the following program?

```
#include <iostream>
int sum(int pt[], int n)
{
   int temp = 0;
   for (int i = 0; i < n; ++i) {
      temp += pt[i];
   }
   return temp;
}

int main()
{
   int total;
   int pt[5];
   for (int i = 0; i < 5; i++)
      pt[i] = i;
   total = sum(pt, 3);
   cout << total << " " << i << endl;
   return 0;
}
```

SOLN: Well, it may not compile, since `i` is not in the scope where it is defined when it is attempted to be output. If there is output it'll look like `3 <random number>`.

11. Which of the following statements contains a reference declarator?

(a) `int *i;`
(b) `swap(&x, &y);`
(c) `int x, *pt; pt = &x;`
(d) `int x, &pt = x;`
(e) None of the above.

SOLN: Only (d) declares a reference variable.

12. What would be printed by the following statements? Why?

```
double x = !0 * (1 / 3 * 3.0);
cout << x << endl;
```

SOLN: Since 1 / 3 is zero, there is a zero factor and so the product is zero.

13. Explain how to dynamically (a) allocate and (b) deallocate a **`float`** array `X[5]` by using **`new`** and **`delete`** operators.

SOLN: You can dynamically allocate an array of 5 elements by, say
`X = new float[5];`
Then you can deallocate the array with
`delete [] X;`

14.
```
int foo(int x)
{
   if(x == 0 || x == 1 || x == 2)
      return x;
   else if( x <= 5)
      return x*x - foo(x - 1);
   else
      return x/2 + foo(x - 1);
}
```

What are the return values for the following function calls? (Hint: trace each function call)

```
foo(1) = 1
foo(2) = 2
foo(3) = 7
foo(4) = 9
foo(5) = 16
foo(6) = 19
foo(7) = 22
foo(8) = 26
foo(9) = 30
```

15. Please complete the following program (by filling in lines 2, 6, 17, 24, 27, 28, 38) so that it will read data from the file `test.dat` and print out the data.

```
1   #include <iostream>
2   #include <fstream>
3
4   #define MAX 10
5
6   class Student {
7   public:
8      char name[25];
9      int Ex1, Ex2;
11  };
11
12  int main()
13  {
14     Student st[MAX];
15     int count = 0;
16
17     ifstream in;
18     if (!in)
19     {
20        cout << "Cannot open
                    test.dat.\n";
21        return 1;
22     }
23
24     in.open("test.dat");
25     while(!in.eof())
26     {
27        in >> st[count].name;
28        in >> st[count].Ex1
             >> st[count].Ex2;
29        cout << st[count].name <<
          " " << st[count].Ex1 <<
30        " " << st[count].Ex2 <<
          "\n";
31        count++;
32        if (count >= 10)
33        {
34           cout << "Exceed MAX: "
                  << MAX << endl;
35           in.close();
36           return(-1);
37        }
38        //no code needed here
39     }
40
41     in.close();
42     return 0;
43 }
```

The input file test.dat contains the following data:

```
Eric 77 87
Scott 90 94
Mary 100 100
```

16. What is the output of the following code?

```
class Nmbr
{
private:
   int m_num;
public:
   Nmbr(int n);
   Nmbr();
   int GetNmbr();
   void SetNmbr(int n);
   Nmbr operator =(Nmbr n2);
   Nmbr operator *(Nmbr n2);
};

Nmbr::Nmbr()
{
   m_num = 0;
}
Nmbr::Nmbr(int n)
{
   m_num = n;
}
int Nmbr::GetNmbr()
{
   return m_num;
}
void Nmbr::SetNmbr(int n)
{
    m_num = n;
}
Nmbr Nmbr::operator *(Nmbr n2)
{
   return Nmbr(m_num * n2.m_num);
}
Nmbr Nmbr::operator =(Nmbr n2)
{
   m_num = n2.m_num * 2;
   return Nmbr(m_num);
}
int main()
{
   Nmbr n1(5), n2(3), n3;
   n3 = n1 * n2;
   cout << n3.GetNmbr() << endl;
   return 0;
}
```

Output: 15