Name: _____

Write all of your responses to the short answer and traces on these exam pages. Use the extra paper provided to write the coding exercises.

1 Short Answer/Function Construction (5 Points Each)

1. What is the difference between passing a parameter by value and passing a parameter by reference? Give an example of two function headers, one that passes an integer parameter x by value and one that passes the parameter x by reference.

2. Make the following declarations:

   (a) A double constant named `PI` with value 3.141659.

   (b) A string initialized with your name.

   (c) A character named `ch` initialized to an uppercase T.

   (d) An integer that stores the ASCII value of a character named `ch`.

   (e) An integer that stores a random number between -10 and 10 inclusively.

3. What data types can be used as the test value for a switch statement?

4. What is function overloading and how does C++ accomplish this? Give an example of two function headers that overload a function.

5. Write a function called `check_input` that will take as parameters two integers (by value), the first is a lower bound and the second is an upper bound. The function should take user input (integers) until the input is between the lower and upper bounds, inclusively. When the input value is in the correct range the function will return the value and while it is not in the correct range the function will display an error and ask for the input again.

6. Write a function that will calculate present value. This is the amount of money you need to invest now to have a target amount in n years. If P is the present value, F is the future value (target amount), r is the annual interest rate in decimal form, and n is the number of years that you plan to let the money sit in the account then the following formula will calculate the present value:

$$P = \frac{F}{(1+r)^n}$$

Write a function named `present_value` that performs this calculation. The function should take F, r, and n as parameters (all by value) and return P.

7. Write a function named `number_of_vowels` that will take in a string as a parameter by value and return the number of vowels in the string. The count must be case insensitive and you may assume that the only vowels are a, e, i, o, and u.

8. Write a function named `is_prime` that will take an integer parameter by value and return a boolean determining if the number is prime or not. Recall that a number is prime if it is only divisible by 1 and itself.

9. Write a function named `formal_name` that will take in a string as a parameter by value (assumed to be a person's name) and return the person's name in formal form. For example: "John Doe" would be changed to "Doe, John".

10. Write function prototypes for the five functions above, `check_input`, `is_prime`, `present_value`, `number_of_vowels`, and `formal_name`.

2 Program Traces (15 Points Each)

1. For the given inputs, write the output of the program.

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

int main() {
    int a, b, c;
    double x, y, z;

    cout << "Input integers: ";
    cin >> a >> b >> c;
    cout << "Input doubles: ";
    cin >> x >> y >> z;

    cout << (a / b) << endl;
    cout << (a % b) << endl;
    cout << pow(a, 2) + pow(b, 2) << endl;
    b++;
    a--;
    c += 3;

    cout << a << " " << b << " " << c << endl;

    c = x;
    y = a++ + c--;
    z = x * y;

    cout << a << " " << b << " " << c << endl;
    cout << x << " " << y << " " << z << endl;

    return 0;
}
```

```
Input integers: 6 4 9
Input doubles: 2.2 3.5 4.7
```

2. For the given inputs, write the output of the program.

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

int fct(int a, int b, int &c, int d) {
    cout << "fct: " << a << " " << b << " " << c << " " << d << endl;
    c++;
    d = a - b;
    a = 2 * a;
    return a;
}

int main() {
    int a, b, c, d;

    cout << "Input integers: ";
    cin >> a >> b >> c;

    int t = 0;
    d = b;
    for (int i = a; i <= b; i += c) {
        t++;
        if (t % 2 == 1) {
            d = fct(i, a, t, c);
        }
        cout << "main: " << a << " " << b << " " << c << " " << d << " " << i
                << " " << t << endl;
    }

    return 0;
}
```

```
Input integers: 3 10 2
```

3 Coding (15 Points Each)

1. Write a program that will bring in a line of text from the user that ends with a period. The sentence will have no other punctuation in it and you are to stop processing the sentence when you hit the period. Have the program count the number of words, count the number of words with one or two letters, count the number of words with 10 or more letters, find the percentages of small and large words (small is 1 or 2 letters and large are those with 10 or more letters), and find the average number of vowels per word.

   Program Run: The user typed in the italicized text.

   `Input some text:` *Far out in the uncharted backwaters of the unfashionable end of the western spiral arm of the Galaxy lies a small unregarded yellow sun.*

   ```
   Average number of vowels = 1.75
   Number of words in the document = 24
   Number of one and two letter words = 5
   Small word percentage = 20.8333%
   Number of words with 10 or more letters = 3
   Large word percentage = 12.5%
   ```

2. Write a function called `factorial` that takes in a single integer parameter by value and returns a long which is the factorial of the parameter. Recall that $n! = 1 \cdot 2 \cdot 3 \cdots n$ when $n > 0$ and we define $0! = 1$. Write a function called `combinations` that will take in two integer parameters by value, n and k and return the number of ways to choose k items from a set of n items. The formula for this is

   $$\frac{n!}{k! \cdot (n-k)!}$$

   Use prototypes for both of these and place the implementation of the functions below the main program. The main will ask the user for an integer, then the program will print out the first n rows of Pascal's Triangle. Note that here line 1 is 0 choose 0, line 2 is 1 choose 0 then 1 choose 1, and so on. The output should look like the example below.

   ```
   Input n: 10
   The first 10 rows of Pascal's Triangle.
       1
       1       1
       1       2       1
       1       3       3       1
       1       4       6       4       1
       1       5      10      10       5       1
       1       6      15      20      15       6       1
       1       7      21      35      35      21       7       1
       1       8      28      56      70      56      28       8       1
       1       9      36      84     126     126      84      36       9       1
       1      10      45     120     210     252     210     120      45      10       1
   ```