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.

   When you pass by value the parameter value is sent to the function and the value is stored in a different spot in memory than it is in the calling function. Hence any change to the variable in the function does not affect the value of the variable in the calling function. When a variable is passed by reference, the memory address of the variable is sent to the function. So the variable in the function and in the calling function share the same memory location. Hence any change to the variable in the function will also change the value in the calling function.

   To pass by reference you place an `&` in front of the parameter name.

   ```
   void fct(int x)
   void fct(int &x)
   ```

2. Make the following declarations:

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

   ```
   const double PI = 3.141659;
   ```

   (b) A string initialized with your name.

   ```
   string myname = "Don";
   ```

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

   ```
   char ch = 'T';
   ```

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

   ```
   int asciich = static_cast<int>(ch);
   ```

   or simply

   ```
   int asciich = ch;
   ```

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

   ```
   int val = rand() % 21 - 10;
   ```

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

   Any integer data type, short, unsigned short, int, unsigned int, long, unsigned long, long long, unsigned long long, char, bool.

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

   Function overloading is when two different functions have the same name. In C++, like Java, the parameter lists must be different.

   ```
   void fct(int x)
   void fct(double x)
   ```

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.

```
int check_input(int lower, int upper) {
    int input = 0;
    do {
        cout << "Input an integer in the range " << lower << " to " << upper
                << ": ";
        cin >> input;
        if (input < lower || input > upper)
            cout << "Invalid input, please try again." << endl;
    } while (input < lower || input > upper);

    return input;
}
```

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.

```
double present_value(double F, double r, double n) {
    return F / pow(1 + r, n);
}
```

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.

```
int number_of_vowels(string str) {
    int count = 0;
    for (unsigned int i = 0; i < str.length(); i++) {
        char ch = str[i];
        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
            ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
            count++;
    }

    return count;
}
```

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.

```
bool is_prime(int num) {
    for (int i = 2; i < num; i++) {
        if (num % i == 0)
            return false;
    }

    return true;
}
```

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".

```
string formal_name(string str) {
    int pos = str.find(" ");
    string first = str.substr(0, pos);
    string last = str.substr(pos + 1);
    return last + ", " + first;
}
```

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

```
int number_of_vowels(string);
int check_input(int, int);
double present_value(double, double, double);
bool is_prime(int);
string formal_name(string);
```

2 Program Traces (15 Points Each)

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

```cpp
#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
```

Solution:

```
1
2
52
5 5 12
6 5 1
2.2 7 15.4
```

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

```cpp
#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
```

Solution:

```
fct: 3 3 1 2
main: 3 10 2 6 3 2
fct: 5 3 3 2
main: 3 10 2 10 5 4
fct: 7 3 5 2
main: 3 10 2 14 7 6
fct: 9 3 7 2
main: 3 10 2 18 9 8
```

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%
```

```cpp
#include <iostream>
#include <string>

using namespace std;

int number_of_vowels(string);

int main() {
    int count = 0;
    string word;
    int sum = 0;
    int countLarge = 0;
    int countSmall = 0;
    bool endOfSentence = false;

    cout << "Input some text: ";

    while (!endOfSentence) {
        cin >> word;

        if (word.find(".") != word.npos) {
            endOfSentence = true;
            word = word.substr(0, word.length() - 1);
        }

        sum += number_of_vowels(word);
        count++;

        if (word.length() <= 2)
            countSmall++;

        if (word.length() >= 10)
            countLarge++;
    }

    double average = 1.0 * sum / count;
    double percentSmall = 100.0 * countSmall / count;
    double percentLarge = 100.0 * countLarge / count;

    cout << "Average number of vowels = " << average << endl;
    cout << "Number of words in the document = " << count << endl;
    cout << "Number of one and two letter words = " << countSmall << endl;
    cout << "Small word percentage = " << percentSmall << "%" << endl;
    cout << "Number of words with 10 or more letters = " << countLarge << endl;
    cout << "Large word percentage = " << percentLarge << "%" << endl;

    return 0;
}

int number_of_vowels(string str) {
    int count = 0;
    for (unsigned int i = 0; i < str.length(); i++) {
        char ch = toupper(str[i]);
        if (ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
            count++;
    }

    return count;
}
```

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
```

```cpp
#include <iostream>
#include <iomanip>

using namespace std;

long factorial(int);
long combinations(int, int);

int main() {
    int n;

    cout << "Input n: ";
    cin >> n;

    cout << "The first " << n << " rows of Pascal's Triangle." << endl;
    for (int j = 0; j <= n; j++) {
        for (int i = 0; i <= j; i++)
            cout << setw(5) << combinations(j, i) << "  ";
        cout << endl;
    }

    return 0;
}

long factorial(int n) {
    long fact = 1;

    for (int i = 1; i <= n; i++)
        fact *= i;

    return fact;
}

long combinations(int n, int k) {
    return factorial(n) / (factorial(k) * factorial(n - k));
}
```