# 1 Short Answer: 3 Points Each

1. Write the line of code that will create a constant named PI and store in it 3.1459.

   **Solution:** `const double PI = 3.14159;`

2. Write s line of code that will set the console output to display exactly two decimal places for all floating point data types, as you would in displaying currency.

   **Solution:** `cout << setprecision(2) << fixed;`

3. Write C++ expressions for the following algebraic expression, $t = \dfrac{x^2 + h}{r^3 - y}$.

   **Solution:** `t = (pow(x, 2) + h)/(pow(r, 3) - y);`

4. What is the value of `!!7` ?

   **Solution:** 1

5. What is "fall through"? In what structure can it occur and what causes it?

   **Solution:** Fall through can happen in a switch statement if a break command is not in one of the cases. If the case is selected the code in that case and all those below it is run until it hits a break or ends the switch statement.

6. The `if` statement regards an expression with a nonzero value as _____.

   **Solution:** True

7. ASCII stands for what? What is it used for and what has it been replaced by?

   **Solution:** American Standard Code for Information Interchange. It is a numeric correspondence to the characters on the keyboard as well as a few other special characters. It has been replaced by unicode which uses it as a subset.

8. Describe the difference between pretest loops and posttest loops. For each of the three types of loops we have discussed list their pretest/posttest types.

   **Solution:** A pretest loop is one that tests the condition before it enters the loop body and a posttest loop is one that checks the condition after the body has run. The for and while are pertest and the do-while is a posttest.

9. What header file do you need to include in a program that performs file operations?

   **Solution:** fstream

10. Which loop should you use in situations where you wish the loop to repeat until the test expression is false, and the loop should not execute if the test expression is false to begin with?

    **Solution:** A while loop.

# 2 Program Traces: 15 Points Each

1. Write the output of the following program.

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

using namespace std;

int main() {
    int x, y, z;

    cout << "Input: ";
    cin >> x >> y >> z;

    if (x > y) {
        cout << "One" << endl;
        if (z < x && 3 * x > y)
            cout << "Two" << endl;
        else
            cout << "Three" << endl;
    } else {
        cout << "Four" << endl;
        if (z < x || z > y)
            cout << "Five" << endl;
        else
            cout << "Six" << endl;
    }
    cout << "-----" << endl;

    x = 4; y = 3; z = 7;
    x = z++ * --y;
    cout << x << " " << y <<
       " " << z << endl;

    string str1 = "Exam 1 is too easy."
        ;
    string str2 = "to";
    string str3 = "eat";

    cout << str1.length() << endl;
    cout << str1.at(10) << endl;
    cout << str1[7] << endl;

    cout << "-----" << endl;

    int pos = str1.find(str2);
    cout << pos << endl;
    cout << str1.substr(pos, 7) <<endl;
    cout << str1.substr(3, pos) <<endl;

    cout << "-----" << endl;
    pos = str1.find_first_of(str3);
    cout << pos << endl;
    pos = str1.find_last_of(str3);
    cout << pos << endl;

    return 0;
}
```

**Output**

```
Input: 5 9 15
```

**Solution:**

```
Four
Five
-----
14 2 8
19
t
i
-----
10
too eas
m 1 is too
-----
2
15
```

2. Write the output of the following program.

```
#include <iostream>

using namespace std;

int main() {
    int x, y, z;

    cout << "Input: ";
    cin >> x >> y >> z;

    for (int i = x; i <= y; i++) {
        cout << i << " ";
        if (i > z) {
            cout << z;
            z -= 2;
        }
        cout << endl;
    }

    cout << "-----" << endl;
    cout << x << " " << y << " " << z
        << endl;
    cout << "-----" << endl;

    while (x > z) {
        cout << x << " " << z << endl;
        x--;
        z++;
    }

    cout << "-----" << endl;
    cout << x << " " << y << " " << z
        << endl;
    cout << "-----" << endl;

    do {
        cout << y << " " << z << endl;
        if (y > z)
            y--;
        else
            y++;
    } while (y != z);

    return 0;
}
```

**Output**

```
Input: 3 9 6
```

**Solution:**

```
3
4
5
6
7 6
8 4
9 2
-----
3 9 0
-----
3 0
2 1
-----
1 9 2
-----
9 2
8 2
7 2
6 2
5 2
4 2
3 2
```

# 3 Programming: 20 Points Each

1. Write a complete C++ program with a loop that lets the user enter a series of integers. The user should enter −99 to signal the end of the series. After all the numbers have been entered, the program should display the largest and smallest numbers entered. A program run is below.

```
Enter Number (-99 to quit): 2
Enter Number (-99 to quit): 6
Enter Number (-99 to quit): 4
Enter Number (-99 to quit): 9
Enter Number (-99 to quit): 5
Enter Number (-99 to quit): -99
Largest = 9
Smallest = 2
```

**Solution:**

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

int main() {
    int largest = 0, smallest = 0, value = 0;

    bool firsttime = true;
    do {
        cout << "Enter Number (-99 to quit): ";
        cin >> value;

        if (value != -99) {
            if (firsttime) {
                firsttime = false;
                largest = value;
                smallest = value;
            }

            if (value > largest)
                largest = value;

            if (value < smallest)
                smallest = value;
        }

    } while (value != -99);

    cout << "Largest = " << largest << endl;
    cout << "Smallest = " << smallest << endl;

    return 0;
}
```

2. Write a complete C++ program that will continually roll a die until a given run length of the same roll happens. The user will input the run length, 1 or greater. Error check the input and reask for input if needed. The program should then simulate rolling the 6-sided die until the same roll comes up the desired number of times consecutively. For example, say the user wants a run of 4. If the simulation is,

```
2 5 4 3 6 4 4 2 1 1 1 3 2 4 5 4 4 4 2 5 3 3 3 3
```

Then the simulation would stop and the count would be 24. The main should set the seed of the random number generator to the time of the system clock. Three runs are below.

```
Input a run length (>= 1): 5
The number of rolls to get 5 consecutive equal rolls was 3482.

Input a run length (>= 1): 10
The number of rolls to get 10 consecutive equal rolls was 18083797.

Input a run length (>= 1): 7
The number of rolls to get 7 consecutive equal rolls was 17787.
```

**Solution:**

```
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main() {
    srand(time(0));
    int count = 1;
    int runlength = 0;

    do {
        cout << "Input a run length (>= 1): ";
        cin >> runlength;

        if (runlength < 1)
            cout << "Invalid input, please input a number >= 1." << endl;
    } while (runlength < 1);

    int runcount = 1;
    int lastroll = rand() % 6 + 1;
    while (runcount < runlength) {
        int die = rand() % 6 + 1;

        if (die == lastroll)
            runcount++;
        else
            runcount = 1;

        lastroll = die;
        count++;
    }

    cout << "The number of rolls to get " << runlength << " consecutive equal
        rolls was " << count << "." << endl;
    return 0;
}
```