COSC 120 **Exam 2 Key** Spring 2012

Part 1: Short Answer (5 Points Each)

1. Write a declaration that creates an array of doubles `arr` and loads it with the values 2.3, 5.72, 3.91 and 4.37. This is to be done in one line all in the declaration.
```
double arr[] = {2.3, 5.72, 3.91, 4.37};
```

2. Give a brief description of what *passed by pointer* means. --- Arrays are passed by pointer, which means that the memory address of the beginning position of the array is passed to a function. It is similar to being passed by reference in that changes to the array are visible in the calling function. One can lock the array from being changed by using the const keyword in the parameter list.

3. What unexpected side effect might happen in the following block of code and why would it happen?

```
int A[10];

for (int j = 0; j < 10; j++)
    cout << A[j] << "  ";
```

   Since the array is not initialized the output will be garbage.

4. Write a typedef statement that will allow the user to use the data type `Henry` in place of an integer array with 29 elements in it.
```
typedef int Henry[29];
```

5. Give a short description of what the keyword `static` does. --- When the keyword static is used on a variable in a function the variable is created and initialized and used as any other variable in the function but it is not destroyed when the function ends. So if the function is called again the value of the variable is retained and can be used without reinitializing it.

6. Write the declaration of a two-dimensional array of longs where the array would have 15 rows and 21 columns.
```
long A[15][21];
```

7. Write a function header for a void function called `DoSomething` that takes in a one-dimensional array of doubles as a parameter and locks out the changing of this array.
```
void DoSomething(const double A[])
```

Part 2: Program Traces (15 points Each)

1. For each of the inputs in the box on the right give the output of the following program.

```
#include <iostream>

using namespace std;

int main()
{
    int A[5];
    int m;

    cin >> m;

    for (int i = 0; i < 5; i++)
        A[i] = m*i;

    for (int i = 0; i < 5; i++)
    {
        int index = ((3*i+1) % m) % 5;
        int temp = A[index];
        A[index] = A[i];
        A[i] = temp;
    }

    for (int i = 0; i < 5; i++)
        cout << A[i] << "  ";

    cout << endl;

    return 0;
}
```

```
2
6  8  2  0  4


3
3  12  0  6  9


5
15  20  10  0  5
```

2. For each of the inputs in the box on the right give the output of the following program.

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

using namespace std;

int main()
{
    int A[5];
    char B[5];
    string str;

    cin >> str;

    for (int i = 0; i < 5; i++)
        cin >> A[i];

    for (int i = 0; i < 5; i++)
        B[i] = str.at(A[i]);

    for (int k = 0; k < 5; k++)
        B[k] = toupper(B[k]);

    for (int k = 0; k < 5; k++)
        if ((B[k] > 'H') && (B[k] < 'R'))
            B[k] = tolower(B[k]);

    for (int i = 0; i < 5; i++)
        cout << B[i];

    cout << endl;

    return 0;
}
```

```
ThomasTheTankEngine
12  7  15  3  10

kHGmA



PleaseTakeANumber
2  5  9  10  1

EEEAl
```

Part 3: Coding (15 points Each)

1. Write a program that will read in integers from a file into an array. The number of integers is unknown but we know that there are less than 2000 of them. Write a value-returning function that will take the array as a parameter and output the number of integers in the array that are larger then 100. Have the main print this number out.

```
#include <iostream>
#include <fstream>

using namespace std;

int countBig(int A[], int size)
{
    int cnt = 0;
    for (int i = 0; i < size; i++)
        if (A[i] > 100)
            cnt++;

    return cnt;
}

int main()
{
    int numbers[2000];
    int arraySize = 0;
    int num = 0;
    ifstream dataFile;
    char ch;

    dataFile.open("numbers001.txt");

    if(!dataFile)
    {
        cout << "Error opening file.\n";
        cin.ignore();
        return 1;
    }

    while ((ch = dataFile.peek()) != EOF)
    {
        dataFile >> num;
        numbers[arraySize] = num;
        arraySize++;
    }
    dataFile.close();

    cout << countBig(numbers, arraySize) << endl;

    return 0;
}
```

2. Below we have the function header for a Binary Search function. Complete the code so that the function will do a binary search on the array of size, size and determine if the target is in the array. If the target is in the array return the index of the target and if the target is not in the array have the function return –1.

```
int BinarySearch(int arr[], int size, int target)
{
    int first = 0;
    int last = size - 1;
    int middle;
    while (first <= last)
    {
        middle = (last + first) / 2;
        if (arr[middle] == target)
            return middle;
        else if (arr[middle] < target)
            last = middle - 1;
        else
            first = middle + 1;
    }
    return -1;
}
```

3. Below we have the function header for a Bubble Sort function. Complete the code so that the function will bubble sort the array of size, size.

```
void BubbleSortArray(int list[], int size)
{
    bool swap;
    int temp;
    int bottom = size - 1;
    do
    {
        swap = false;
        for (int count = 0; count < bottom; count++)
        {
            if (list[count] > list[count+1])
            {
                temp = list[count];
                list[count] = list[count+1];
                list[count+1] = temp;
                swap = true;
            }
        }
        bottom--;
    }
    while(swap != false);
}
```