Sample C++ Exam

1. You are going to create a dynamic array of 15 elements. You are going to write a procedure to read it in, a procedure to write it out, a third one to determine the minimum and maximum.

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

int read(int x[]);
void write(int x[], int n);
void findMinMax(int x, int n, int &min, int &max);

int main(void) {
    int array;
    int i, numValues, min, max;

    array = new int[10];
    numValues = read(array);
    write(array, numValues);
    findMinMax(array, numValues, min, max);
    cout << "Minimum = " << min << " Maximum = " << max << endl;
    return (0);
}

int read(int x[], int n) {
    for (n = 0; n < 10; n++) {
        cout << "Enter a value\t?";
        cin >> x[n];
    }
    return(n);
}

void write (int x[], int n) {
    int i;
    for (i = 0; i < n; i++)
        cout << "x[" << i << "] = " << x[i] << endl;
}

void findMinMax(int x[], int n, int &min, int &max) {
    int i;
    min = max = x[0]; // a set with one value is both
    // the max and min of the set
    for (i = 1; i < n; i++) {
        if (x[i] > max)
            max = x[i];
        if (x[i] < min)
            min = x[i];
    }
    return;
}
```
for (i = 1; i < n; i++) {
    if (x[i] < min)
        min = x[i];
    else if (x[i] > max)
        max = x[i];
}

2. Crate a class that has an array of ten elements and the size, where there are two constructors (a default constructor and a conversion constructor). There is also an accessor called getSize().

class ArrayType {
public:
    ArrayType();
    ArrayType(int newSize);
    int getSize();
private:
    int *array;
    int size;
}

ArrayType::ArrayType(void) {
    array = new int[10];
    size = 10;
}

ArrayType::ArrayType(int newSize) {
    array = new int[newSize];
    size = newSize;
}

int ArrayType::getSize(void) {
    return size;
}

3. What will the following program print?

#include <iostream>
using namespace std;

int main(void) {
    int x = 45, y = 13;
    cout << "x = " << x << "\ty = " << y << endl;
The output is:
  x = 45  y = 13
  a = 45  b = 13
  a = 23  b = 17
  x = 23  y = 13 x is a reference parameter, so main “knows” about the change. This is not true of y because it is a value parameter (the default).

2. Write a class definition called MyStuff that include the following:
   a. the private data items are two double values called x and y and an int z.
   b. the public methods are a default constructor that sets x, y and z equal to zero, a conversion constructor, a copy constructor, a method findAverage which returns their average, an input method read and an output method write.

The complete class, together with a driver main method to run it.

```cpp
#include <iostream>

using namespace std;

class MyStuff   {
public:
    MyStuff(void);
    MyStuff(double newX, double newY, int newZ);
    MyStuff(MyStuff &other);// copy constructor -
    // not on test
```
double findAverage(void);
void read(void);
void write(void);
private:
  double x, y;
  int z;
};

MyStuff::MyStuff(void) {
x = y = z = 0;
}

MyStuff::MyStuff(double newX, double newY, int newZ) {
x = newX;
y = newY;
z = newZ;
}

MyStuff::MyStuff(MyStuff &other) {
x = other.x;
y = other.y;
z = other.z;
}

double MyStuff::findAverage(void) {
  return((x + y + z)/3.0);
}

void MyStuff::read(void) {
cout << "Enter x\t?";
cin >> x;
cout << "Enter y\t?";
cin >> y;
cout << "Enter z\t?";
cin >> z;
}

void MyStuff::write(void) {
cout << "x = " << x << "\ty = " << y << "\tz = " << z << endl;
}

int main(void) {

MyStuff a, b(3.4, 4.5, 6);
MyStuff c(a);

a.write();
a.read();
a.write();
b.write();
c.write();

return(0);

3. Write a program that opens a text file called test.data, reads a single int value, multiplies it by 2 and prints the doubled value on the screen and then closes the file. Include all necessary declarations and methods. This is NOT on the exam, but may be on the final:

```cpp
#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;

int main(void) {
    ifstream infile;
    int x;
    infile.open("test.data");
    if (!infile) {
        cerr << "Cannot open test.data\n";
        exit(1);
    }
    infile >> x;
    x *= 2;
    cout << "The new value is " << x;
    return(0);
}
```