Name: ____

Write all of your responses on these exam pages. If you need extra space please use the backs of the pages. The Short Answer questions are worth 5 points each. The Coding questions are worth 25 points each.

1 Short Answer

1. What is the difference between a protected class member and a private class member?

2. Which constructor is called first, that of the derived class or the base class?

3. When does static binding take place? When does dynamic binding take place?

4. What is an abstract base class? How is it crated? What distinguishes it from a non-abstract base class?

5. Write the first line of the declaration for a Poodle class. The class should be derived from the Dog class with public base class access.

6. Write the first line of the declaration for a SoundSystem class. Use multiple inheritance to base the class on the MP3player class, the Tuner class, and the CDPlayer class. Use public base class access in all cases.

7. What is a container class?

8. What is an iterator? Write the definition of one for a vector of doubles.

9. Write a function named `fct` that takes two integer parameters a and b. If a is less than b the function should throw an exception of an integer error code of 123. Also write the segment of code in the main that will call this function and catch the exception. Output from these segments of code is below.

```
Enter two numbers: 5 3
The sum is 8

Enter two numbers: 1 4
Error 123
```

10. Write a templated function called `print` that brings in a vector of any type and prints its contents on a single horizontal line with a space between the entries. Have it do a new line at the very end and do the printing in a range-based loop.

11. Write a templated bubble, insertion, or selection sort (just one of these) that will sort an array (regular array, not STL) of any data type that has the relational operators < and > defined as well as assignment.

2 Coding Exercise #1

This exercise is to write an inheritance structure for triangles. Recall from geometry that an Isosceles triangle is one with two equal sides and that an Equilateral triangle is one where all three sides are of equal length.

Write a Triangle class that has a default constructor and one that takes in the three sides as parameters. This class should store the lengths of the three sides of the triangle. In addition it should have the following functions.

- Area: this function will calculate and return the area of the triangle. Recall that if the sides of the triangle are a, b, and c then let p be the semi-perimeter $p = (a + b + c)/2$ then the area is
$$A = \sqrt{p(p-a)(p-b)(p-c)}$$
- Sides: this prints to the screen the lengths of the sides.
- Draw: this prints “Draw Triangle” to the screen.

Write an Isosceles class that inherits off the Triangle class. There must be a default constructor and one that takes two parameters, the first is the double sides of equal length and the third is the length of the last side. That is, `Isosceles(3, 5)` is a triangle with side lengths 3, 3, and 5. This class must also be able to call the functions Area, Sides, and Draw, but in this case the Draw command will print to the screen, “Draw Isosceles Triangle”.

Write an Equilateral class that inherits off the Isosceles class. There must be a default constructor and one that takes one parameter, the length of all the sides. That is, `Equilateral(7)` is a triangle with side lengths 7, 7, and 7. This class must also be able to call the functions Area, Sides, and Draw, but in this case the Draw command will print to the screen, “Draw Equilateral Triangle”.

Write the specifications and implementations for each of the three classes below. **Each of the following pages is for each separate file, the filename to be written is at the top of the page.**

There is to be no inline code. There is a block of sample code below and its output. Read this very closely, your class structures are to produce exactly the same output to this sample code.

Sample Code

```
Triangle *tris[5];
tris[0] = new Triangle(3, 4, 5);
tris[1] = new Isosceles(3, 5);
tris[2] = new Isosceles(4, 5);
tris[3] = new Equilateral(7);
tris[4] = new Triangle(7, 5, 3);

for (int i = 0; i < 5; i++) {
    tris[i]->Draw();
    cout << tris[i]->Area() << endl;
    tris[i]->Sides();
    cout << endl;
}
```

Output

```
Draw Triangle
6
Side Lengths: 3 4 5

Draw Isosceles Triangle
4.14578
Side Lengths: 3 3 5

Draw Isosceles Triangle
7.80625
Side Lengths: 4 4 5

Draw Equilateral Triangle
21.2176
Side Lengths: 7 7 7

Draw Triangle
6.49519
Side Lengths: 7 5 3
```

Triangle.h

Triangle.cpp

Isosceles.h

Isosceles.cpp

Equilateral.h

Equilateral.cpp

3 Coding Exercise #2

Write a templated class named `Tarray` that stores a dynamic one-dimensional array (normal array, not an STL array) of the parameter type T. It also stores the size of the array and a default value for populating the array. The class should have the following functions. Example main and output is below. There is to be no inline code. **Each of the following pages is for the code of the specified functions.**

- Constructor that brings in parameters for the array size and default value. No default vales here and no default constructor is to be written.
- Destructor, copy constructor and overloaded assignment operator.
- A size function that returns the size if the array.
- Min and Max functions that return the minimum and maximum values in the array.
- Set function that takes in an array index and values and places that value in the array. If the index is out of bounds of the allocated array memory then the function should throw an exception string of `Index out of bounds.`.
- Get function that returns the element of the array at the specified (parameter) index. If the index is out of bounds of the allocated array memory then the function should throw an exception string of `Index out of bounds.`.
- Overloaded [ ] operator. If the index is out of bounds of the allocated array memory then the function should throw an exception string of `Index out of bounds.`.
- A print function that prints the array elements with a comma between the entries.

```
#include "Tarray.h"
#include <iostream>
using namespace std;

int main() {
  Tarray<int> A(10, 0);
  A.Print();
  for (int i = 0; i < A.size(); i++)
    A[i] = (3 * i + 1) % A.size();
  A.Print();
  cout << A.Max() << " " << A.Min() <<
      endl;
  cout << endl;
  Tarray<int> B(5, 3);
  Tarray<int> C(20, 1);
  B.Print();
  C.Print();
  cout << endl;
  B = C = A;
  A.Print();
  B.Print();
  C.Print();
  cout << endl;
  B[2] = B[4] = B[7] = -5;
  C[1] = C[2] = C[3] = -1;
  A.Print();
  B.Print();
  C.Print();
  cout << endl;
  Tarray<int> D(A);
  A.Print();
  D.Print();
  D.set(5, -2);
  D.set(1, -5);
  D.set(7, 0);
  D.Print();
  for (int i = 0; i < A.size(); i++)
    cout << A.get(i) << " ";
  cout << endl << endl;
  Tarray<string> S(5, "Jack");
  S[0] = "Sue";
  S[1] = "James";
  S[2] = "George";
  S[3] = "Don";
  S.Print();
  cout << S.Max() << " " << S.Min() <<
      endl;
  return 0;
}
```

Output:

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

3, 3, 3, 3, 3
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1

1, 4, 7, 0, 3, 6, 9, 2, 5, 8
1, 4, 7, 0, 3, 6, 9, 2, 5, 8
1, 4, 7, 0, 3, 6, 9, 2, 5, 8

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

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

Sue, James, George, Don, Jack
Sue Don
```

Tarray Specification: You may assume that all needed includes are included, the specification is guarded, and that you are using the std namespace.

Constructor and Destructor

Copy Constructor and overloaded =

Size, Min, Max, and Print

Get, set, and overloaded [ ]