Exam 2

PRACTICE

1. True or False

a) Multi-dimensional arrays are only possible with integral data types (e.g., bool, integer, const).

b) All getter functions are void functions.

c) Non-member functions have access to public members of an object.

d) A function in main() has the same access to a member function as other member functions of the class.

2.Code

gnome.h

```
class Gnome {

   public:
      Gnome();
      Gnome(int, int);
      int GetVal1() const;
      int GetVal2() const;

   private:
      int value1;
      int value2;

};
```

main.cpp

```
int main()  {
   Gnome a(10,25);
   cout << a.value1 << “ ” << a.value2;
   cout << endl;
   return 0;
}
```

2.Questions

a) What type of message would the compiler display?

b) Correctly rewrite the line of code to correct the error.

c) What is the purpose of const in the two member functions?

d) What is Gnome(), and why isn’t there a return type?

3.Code: what is legal?

Given this class definition and main function, what statements in main are legal?

```
class Gnome {

   public:
               Gnome();
               Gnome(int v1, int v2);
               int GetValue1() const;
               int GetValue2() const;
   private:
               int value1;
               int value2;

};
```

```
int main() {

   Gnome g1;
   Gnome g2();
   g1.value1 = 52;
   int value1;
   value1 = g1.GetValue1();
   Gnome g3 = g1;
   g3.g2();
   cout << value1;
   cout << value2;

}
```

4.Short Answer

a) Suppose you have developed a class called MyClass. Write the function header for this class's default constructor.

b) How many elements are in myVec after the following declaration:

```
vector<int> myVec;
```

c) Suppose the first element of an integer array is at base address 550. What is the base address of the 4th element, i.e., myArray[3]? (Note: each integer takes four bytes.)

5.Functions

Consider the following function:

```
Circle Circle::DoSomething(const Circle& c) {
      // does something here
}
```

a) What is the name of the function?

b) Is this function a member function? If yes, to what class

6. Functions … cont.

Consider the following function:

```
Circle Circle::DoSomething(const Circle& c) {
      // does something here
}
```

a) What does the first Circle represent?

b) What does the third Circle represent?

7. File I/O … Find the errors

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

int main () {
    fout fstream;
    double x(2.0), y(2.5);
    if (x > y)
        fout >> “x is greater than y” << endl;
    else
        fout >> “x is less than/equal to y” << endl;
    return 0;
}
```

8. File I/O … Write Code

Write snippet of code that (a) declares and opens the file “FileIn.txt” for an input stream named “myInput” and (b) checks to be sure the open occurred and, if not, exit the program.

Suppose FileIn.txt contains 3 integers. Write code to read in these three integers from the input stream named “myInput”.

9. Constructors

Which of the following are valid constructors. Justify issue, if one exists.

a) BankAccount::BankAccount() const;

b) BankAccount::BankAccount(double balance);

c) void BankAccount::BankAccount();

d) BankAccount::BankAccount(const string& acct, double balance);

10. Member Functions

Which of the following are valid member functions. Justify issue, if one exists.

a) double HotDog::GetPrice( ) const

b) Triangle::CalculateArea( )

c) Buffalo Buffalo::buffalo(Buffalo buffalo)

d) void Dog::FetchBall

e) double AlarmClock::Ring(float)

11. What is printed?

```
class Gnome {

   public:
      Gnome();
      Gnome(int v1, int v2);
      int GetVal1() const;
      int GetVal2() const;
      int Diff();
      int Diff(const Gnome& g);

   private:
      int value1;
      int value2;

};
```

```
int Gnome::Diff() {
    return (value2 - value1);
}
int Gnome::Diff(const Gnome& g) {
    return (value2 - g.value2);
}
```

```
int main()
{
    Gnome a(10,25), b(5,20);
    cout << a.Diff() << endl;
    cout << a.Diff(b) << endl;
}
```

12. Drawing Time

Draw a picture of the array that would be created by the following code.

```
int data[ 8 ] = { 1, 1 };

for( int index = 2; index < 8; ++index )  {
          data[ index ] = data[ index – 1 ] + data[ index – 2 ];
}
```

13. Sequential Search

Suppose you have the following array defined:

```
const int NROWS = 4;
const int NCOLS = 2;
char data[ NROWS ][ NCOLS ] = { ‘9’, ‘2’, ‘8’, ‘5’, ‘1’, ‘3’, ‘4’, ‘8’};
```

Write the code to search whether the character ‘3’ is in the 2D array.

14. Composition Questions

```
class Swim {   // in Swim.h
   public:
      const static double DIST = 0.75;
      Swim();
      // all get and set functions
   private:
      double time;
   };
   class Run {   // in Run.h
   public:
      const static double DIST = 20.0;
      Run();
      // all get and set functions
   private:
      double time;
   };
class Bike {   // in Bike.h
   public:
      const static double DIST = 5.0;
      Bike();
      // all get and set functions
   private:
      double time;
   };
```

Write the code for the following statements (no need to include the main() function).

a) Declare a Triathlon name theTriathlon.

b) Set theTriathlon’s race name to Colorado Sprint.

c) Set the competitor’s time for the Bike portion of the race to 37 and then print this time.

d) Display the total distance a competitor covers in theTriathlon using defined constants.

15. Classes: Put it all together

Write the class declaration for a new data type called **Point**. Use the keyword **const** where it is appropriate to do so. Your Point data type should have:

a. two double data members (x and y) that must always be positive
b. two constructor functions, one default and one that takes both x and y
c. getter/setter functions for each data member
d. one member function, called Distance, that takes one parameter and returns the distance between two point objects, and
e. one private helper function, called Check, that returns a boolean on whether a point has positive x and y. (All non-default constructors set x,y and then call this function. If false, then set x,y to the default values.)

16. Classes: Put it all together

a. Write the function implementation of your parameterized constructor for your Point data type.

b. Write the function implementation for one of your setter functions in your Point data type.

c. Write the function implementation for the member distance function of your Point data type. Assume sqrt() function is available to you (via the <cmath> header file.

d. Write the helper function for the Point data type.

17. Classes: Put it all together

Write a main function that:

a). declares two Point objects that are initialized with (5,3) and (7,1)
b). prints the values of the (5,3) point using the accessor functions
c). changes the value of the 1 in the (7,1) point to -3
d). prints the distance between the two points
e). prints whether the (7,-3) point is a valid Point using Check()

18. Army of Gnomes?

```
class Gnome {

  public:
    Gnome();
    Gnome(int v1, string name);
    int GetVal1() const;
    string GetName() const;

  private:
    int value1;
    string name;

};
```

Declare a vector of Gnomes.
Then add two Gnomes:
Harry with value 35
Sally with value 33

```
//Army of Gnomes?
int main()
{



}
```

19. Composition

```
class Chair {    // defined in Chair.h
   private:
      int height, width, depth;
      double price;
   public:
       const static int DIMENSION = 1;
       Chair();
       Chair(int, int, int, double);
       // All get and set functions
};

class Table {      // defined in Table.h
   private:
      int height, width;
     double price;
   public:
     Table();
     Table(int, int, double);
     // All get and set functions
};
```

Write a .h file to define a new object type: DiningSet.
DiningSet has 2 chairs and 1 table, a bool on whether set is sold, and a GetPrice() function.

```
//Dining Set with 2 chairs and 1 table?
```

20. Composition: Put it all together

a. Write the function implementation of the Chair's default constructor. Use 10.0 for the price, and DIMENSION for height, width, and depth.

b. Write a statement to print DIMENSION to the screen in main().

c. Write the implementation of GetPrice() in your DiningSet class. GetPrice() is equal to the sum of the table's price plus the two chairs' prices.