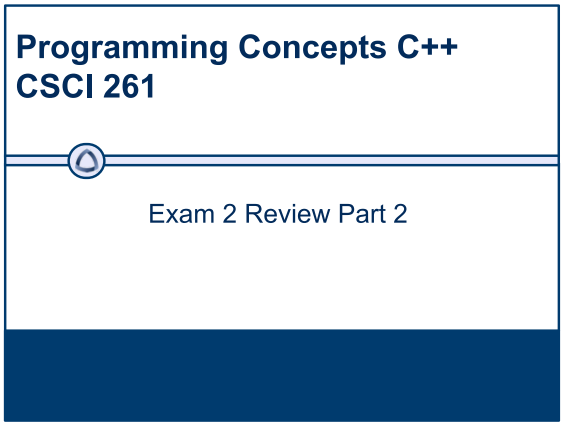
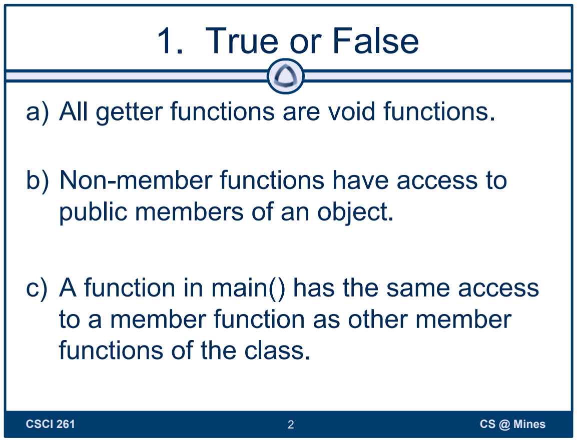


Programming Concepts C++
CSCI 261

Exam 2 Review Part 2



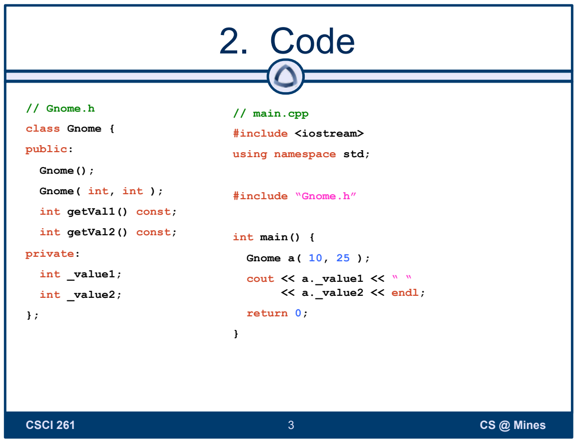
1. True or False

a) All getter functions are void functions.

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

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

CSCI 261 2 CS @ Mines

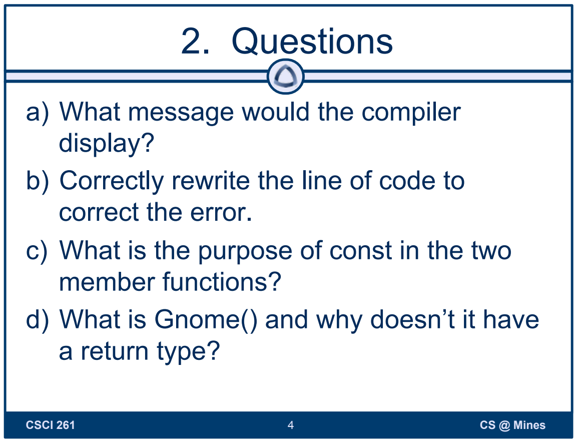


2. Code

```
// Gnome.h
class Gnome {
public:
  Gnome();
  Gnome( int, int );
  int getVal1() const;
  int getVal2() const;
private:
  int _value1;
  int _value2;
};
```

```
// main.cpp
#include <iostream>
using namespace std;


#include "Gnome.h"


int main() {
  Gnome a( 10, 25 );
  cout << a._value1 << " "
       << a._value2 << endl;
  return 0;
}
```

CSCI 261 3 CS @ Mines



2. Questions

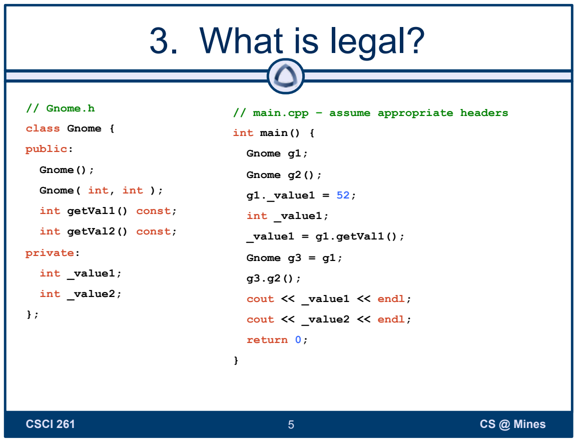
a) What message would the compiler display?

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 doesn't it have a return type?

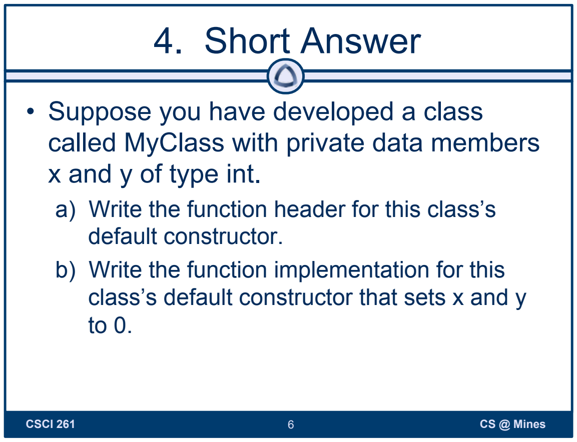
CSCI 261 4 CS @ Mines



3. What is legal?

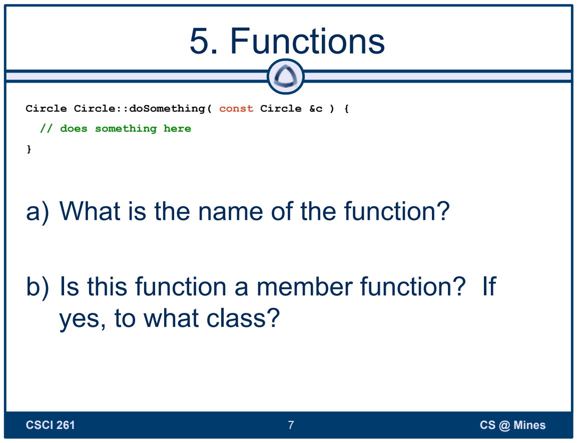
```
// Gnome.h
class Gnome {
public:
  Gnome();
  Gnome( int, int );
  int getVal1() const;
  int getVal2() const;
private:
  int _value1;
  int _value2;
};
```

```
// main.cpp - assume appropriate headers
int main() {
  Gnome g1;
  Gnome g2();
  g1._value1 = 52;
  int _value1;
  _value1 = g1.getVal1();
  Gnome g3 = g1;
  g3.g2();
  cout << _value1 << endl;
  cout << _value2 << endl;
  return 0;
}
```



4. Short Answer

- Suppose you have developed a class called MyClass with private data members x and y of type int.
  a) Write the function header for this class's default constructor.
  b) Write the function implementation for this class's default constructor that sets x and y to 0.

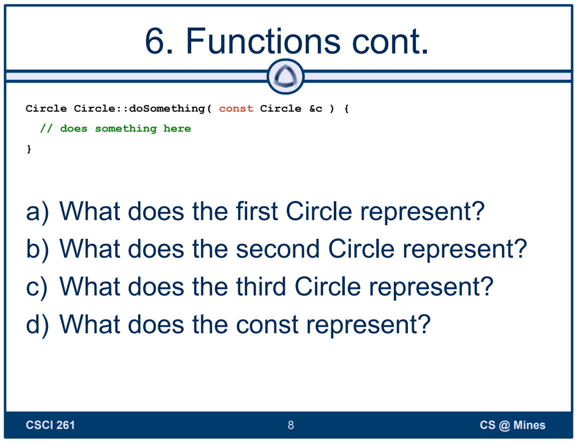


5. Functions

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

a) What is the name of the function?

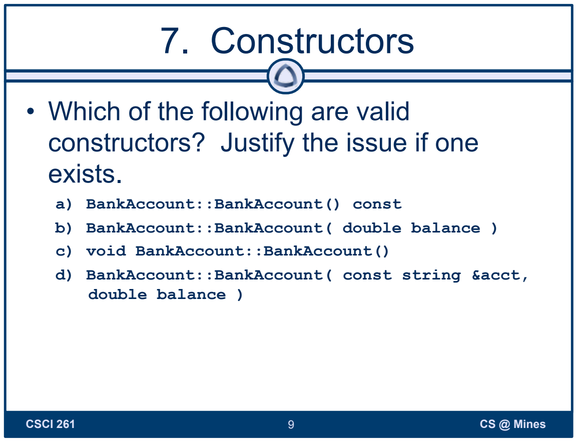
b) Is this function a member function? If yes, to what class?



6. Functions cont.

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

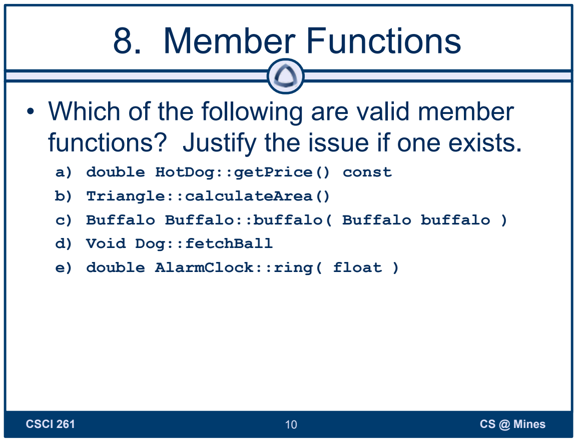
a) What does the first Circle represent?
b) What does the second Circle represent?
c) What does the third Circle represent?
d) What does the const represent?



7. Constructors

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

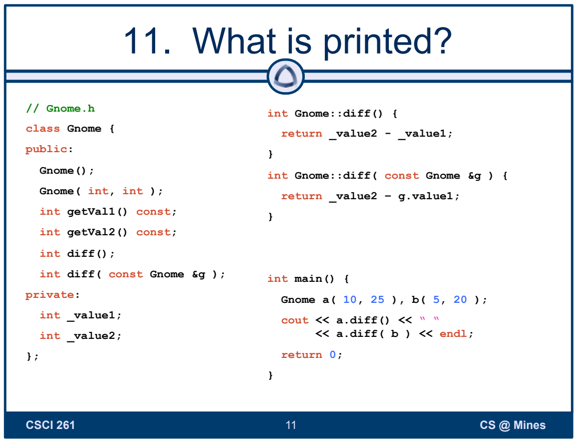
```
a) BankAccount::BankAccount() const
b) BankAccount::BankAccount( double balance )
c) void BankAccount::BankAccount()
d) BankAccount::BankAccount( const string &acct,
    double balance )
```



8. Member Functions

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

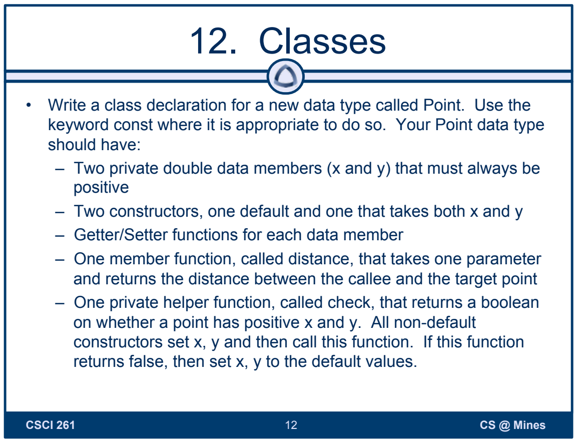
```
a) double HotDog::getPrice() const
b) Triangle::calculateArea()
c) Buffalo Buffalo::buffalo( Buffalo buffalo )
d) Void Dog::fetchBall
e) double AlarmClock::ring( float )
```



11. What is printed?

```
// Gnome.h
class Gnome {
public:
  Gnome();
  Gnome( int, int );
  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.value1;
}


int main() {
  Gnome a( 10, 25 ), b( 5, 20 );
  cout << a.diff() << " "
       << a.diff( b ) << endl;
  return 0;
}
```



12. Classes

- Write a 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:
  - Two private double data members (x and y) that must always be positive
  - Two constructors, one default and one that takes both x and y
  - Getter/Setter functions for each data member
  - One member function, called distance, that takes one parameter and returns the distance between the callee and the target point
  - 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 this function returns false, then set x, y to the default values.

13. Classes cont.

a) Write the function implementation of your parameterized constructor for your Point class

b) Write the function implementation for one of your setter functions in your Point class

c) Write the function implementation for the distance function of your Point class

d) Write the helper function for your Point class

14. Classes cont. cont.

- Write a main function that
  - Declares two Point objects that are initialized to (5, 3) and (7, 1)
  - Prints the values of the (5, 3) point using the accessor functions
  - Changes the value of 1 in the (7,1) point to -3
  - Prints the distance between the two points

15. Army of Gnomes!

```
// Gnome.h
class Gnome {
public:
  Gnome();
  Gnome( int, string );
  int getVal1() const;
  string getName() const;
private:
  int _value1;
  string _name;
};
```

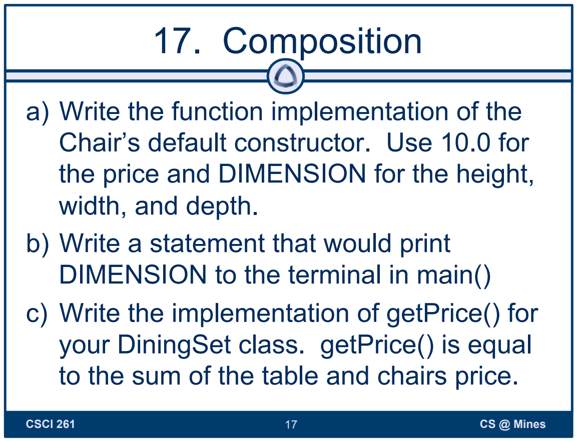
- Declare a vector of Gnomes. Then add two Gnomes:
  - harry with value 35
  - sally with value 38

```
int main() {



}
```

16. Composition

```
class Chair { // in Chair.h
public:
  Chair();
  Chair( int, int, int, double );
  static const int DIMENSION = 1;
  // all getters and setters
private:
  int _height, _width, _depth;
  double _price;
};
class Table { // in Table.h
public:
  Table();
  Table( int, int, int, double );
  // all getters and setters
private:
  int _height, _width, _depth;
  double _price;
};
```

- Write a .h file to define a new class DiningSet. DiningSet has two chairs and one table, a bool on whether the set is sold, and a getPrice() function.



17. Composition

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

b) Write a statement that would print DIMENSION to the terminal in main()

c) Write the implementation of getPrice() for your DiningSet class. getPrice() is equal to the sum of the table and chairs price.