EECE.3220: Data Structures

Spring 2017

Lecture 11: Key Questions
February 10, 2017

1. (Review) Explain what classes and objects are.

2. (Review) Explain what data members are and how they are typically accessed.

3. (Review) Show the general syntax of a class definition and explain the difference between private and public members of a class.

4. Explain the example code below, which provides the class definition and function definitions for a simple class with one data member.

GradeBook.h

```
// GradeBook class interface
class GradeBook
{
public:
   // function that sets the course name
   void setCourseName( string name );

   // function that gets the course name
   string getCourseName();

   // function that displays a welcome message
   void displayMessage();

private:
   string courseName; // course name for this GradeBook
};
```

GradeBook.cpp

```
// GradeBook class implementation
#include "GradeBook.h"

// function that sets the course name
void GradeBook::setCourseName( string name ) {
      courseName = name;
}

// function that gets the course name
string GradeBook::getCourseName() {
      return courseName;
}

// function that displays a welcome message
void GradeBook::displayMessage() {
      cout << "Welcome to the grade book for\n" << courseName
          << "!" << endl;
}
```

5. The example program below shows how objects are declared and their member functions are called. Use the space below the program for notes on these topics.

```
int main()
{
   string nameOfCourse;  // string of chars to store course name
   GradeBook myGradeBook; // new GradeBook object

   // display initial value of courseName
   cout << "Initial course name is: "
          << myGradeBook.getCourseName() << endl;

   // prompt for, input and set course name
   cout << "\nPlease enter the course name:" << endl;
   getline( cin, nameOfCourse ); // read course name with blanks

   myGradeBook.setCourseName( nameOfCourse );
   cout << endl;
   myGradeBook.displayMessage();
   return 0;
}
```

6. Explain the purpose of constructors.

7. Explain the difference between default and parameterized constructors. Write the default and parameterized constructors for the GradeBook class.

8. Explain function overloading in C++.

9. Assume that GradeBook.h is as follows:

```
#include <string>
using std::string;
class GradeBook
{
public:
    GradeBook( );
    GradeBook( string name );
    void setCourseName(string name);
  string getCourseName();
    void displayMessage();
private:
   string name;
}; // end class GradeBook
```

Which of the following statements would be legal in a main program that uses the GradeBook class? Which would cause compiler errors? What code could we use to fix those errors?

a. `GradeBook g1(3220);`

b. `GradeBook g2;`

c. `setCourseName(g2);`

d. `g2.courseName = "EECE.3220";`

e. `string s = g2.getCourseName();`

f. `g2.displayMessage;`