**LESSON SET 13**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Introduction to Classes**

This lab briefly introduces the student to object-oriented programming and the

concept of classes and objects in C++.

**OBJECTIVES FOR STUDENT**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Lesson 13A:**

1. To declare a class and define objects of an existing class

2. To call member functions of a class

3. To implement member functions of a class

4. To declare and use constructors and destructors

5. To develop a client code that makes use of a class

**Lesson 13B:**

6. To use arrays as data members

7. To define and use array of objects

**ASSUMPTIONS**

**Lesson 13A:**

1. Students have read and have a basic understanding of pre-lab reading

assignment

2. Students understand basic concepts of classes and objects

3. Students understand constructors and destructors

**Lesson 13B:**

1. Students understand how to create and use arrays as data members
2. Students understand how to define and use arrays of objects

**PRE-LAB WRITING ASSIGNMENT SOLUTIONS**

1. constructor

2. same

3. methods

4. destructor

5. dot

6. default

7. tilde (~)

8. inline

9. every

10. implicitly (automatically)

**LAB ASSIGNMENTS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Lesson 13A:**

Lab 13.1: Squares as a class

Lab 13.2: The Circles Class

**Lesson 13B**

Lab 13.3: Arrays as data members of classes

Lab 13.4: Array of objects

Lab 13.5 Student generated code assignments

**LESSON 13A**

**LAB 13.1: Squares as a class**

Lab 13.1 asks the students to declare a class called Square. The student should know the member functions and member data by looking at the implementation section which is already completed. The lab also asks the students to fill in client code that uses an object called box defined from the Square class.

A solved program is found in squareKey.cpp in the instructor’s folder for Lesson Set 13.

**Lab 13.2: The Circles Class**

Lab 13.2 asks the student to complete and expand a program that uses objects defined as Circles class. It asks the students to implement the non-default constructor and two other member functions. It then asks the students to alter the code so that the setting of both the center and the radius can be done during the object definition. This is an ambitious lab in that it requires students to develop several constructors of the same class that have different parameters. The problem some students may have is how to have some private data members initialized by the user while others are assigned the default values. Once this concept is realized, the students should proceed through the lab without much difficulty.

A solved program is found in circleKey.cpp in the instructor’s folder for Lesson Set 13.

**LESSON 13B**

**LAB 13.3: Arrays as Data Members**

This program has students working with arrays as data members. It does not

add any new concepts; however, it gives students practice in working with member

function implementation and calls. The program floatarray.cpp reads floating

point data from a data file and places those values into the private data

member called *values* (a floating point array) of the Floatlist class. Those values

are then printed to the screen. The input is done by a member function

called getList, the output is done by a member function called printList. The

amount of data read in is stored in the private data member called length. The

member function getList is called first so that length can be initialized to zero.

The students are asked to fill in code to call the various member functions as well

as write the code for those functions including the constructor and destructor. Some students may find the implementation of the functions a challenge.

The answer to Exercise 1 is as follows:

The function printList includes the word const after its name because it does NOT alter any private member data. The word const is an assurance that the function will not alter private member data. The member function getList does alter private member data and thus does not have the word const after its name.

The solved program is found in floatarrayKey.cpp in the instructor’s folder for Lesson Set 13.

**Lab 13.4 Array of Objects**

This lab introduces array of objects and it requires an understanding of files. The

program documentation is given in the program itself. Students may have some

difficulty in developing the code to bring in the data (inventory id numbers and

number of items) from a file into the array of objects. Although the instructions

give some hints, they do not specifically lay out the format for them.

A complete solution to this lab is given in InventoryKey.cpp in the instructor’s

folder for Lesson Set 13.

**LAB 13.5: Student Generated Code Assignments**

Exercise 1 has the student developing a complete program using a class. It asks

students to develop a SavingsAccount class that allows user to input initial values

of dollars and cents and then asks for deposits and withdrawals. This is a somewhat

complex problem because of normalization. Students must normalize in

most of the member functions. This can be quite a challenge (particularly in

withdrawals). Exercise 2 expands the program by having the student replace the intial member function by two constructors.

A complete solution to this lab is given in bankconKey.cpp in the instructor’s

folder for Lesson Set 13.