Java Programming

2-1: Working with Pre-Written Code

Practice Activities

Lesson Objectives:

- Read and understand a pre-written java program consisting of classes and interacting objects
- Apply the concept of inheritance in the solutions of problems
- Test classes in isolation
- Describe when it is more appropriate to use an ArrayList than an Array

Vocabulary:

Identify the vocabulary word for each definition below.

	A method that can modify an object.
	A method that can access the contents of an object but does not modify that object.
	Object that can store multiple object types and can grow and shrink dynamically as required.
	The process where one object acquires the properties of another.
	Allows you to check the quality of the code for a class independent of the rest of the program code.

Try It/Solve It:

1. If you did not install the JavaBank Case Study during the lesson then please follow the slides from Slide 6 to do so now.

2. Explore JavaBank. Record your observations.

   What happens when you:
   - Display Accounts
   - Create Accounts
   - Delete Accounts
   - Make a Withdrawal Transaction
   - Make a Deposit Transaction
   - Can you display accounts before any are created?
   - Can you create an account without entering anything in the fields?
   - Can you make a withdrawal transaction with no amount in the Withdrawal field?
   - Can you do a deposit transaction with no amount in the Deposit field?
   - What other questions do you have about the JavaBank application?

- What changes would you make to the current application to make it function better?
- What additions would you make to the current application to increase its functionality?

3. Import the file **bikeproject.zip** to a location on your computer and answer the following questions.

    a. Give an example of a primitive data type that is used to store fields within a class.

    b. Give an example of where String concatenation takes place.

    c. What are the names of the objects created in this program?

    d. How many constructors does each class have?

    e. Inheritance is part of this program. Identify the Super and subclasses from this program.

    f. Mountain bikes and road bikes can be constructer either by using the default values (standard bike) or customized to the client's needs. Using the following table identify sample values assigned to one of each type of standard bike:

Values	**Bike (MB)**	**Mountain Bike**	**Bike (RB)**	**Road Bike**
handleBars	Bull Horn		Drop	
frame				
tyres				
seatType				
NumGears				
make				
suspension		RockShox XC32		
type				
frameSize				
tyreWidth				20
postHeight				

4. Working with the Calculator program.
    a. Download, and then unzip the Calculator.zip file from this lesson’s Reference Materials.
    b. From Eclipse, import the Calculator.jar file:
        i. From the File Menu, Select Import
        ii. Expand General
        iii. Select Existing Projects into Workspace
        iv. Click Next
        v. Choose “Select Archive File”
        vi. Click Browse, go to the location of and select the Calculator jar. file, and then click Open
        vii. Click Finish.
    c. Once imported – run the application (CalcMain is the driver).
    d. Determine what Calculator does and how it works – investigate.
    e. Add multiplication and subtraction buttons to the application.
    f. Test to make sure all functionality works as you expect.
    g. Export updated Calculator to a “Runnable” JAR file.
    h. Go to the location where you put the runnable JAR and double click it to run the application.