PACE: African-American and Latino High-School Students Working Together to Excel in Math, Science and Engineering.

Object Oriented Overview (preview for Java programming). Invoking objects from other objects. Objects has properties and behaviors.

Eduardo Pinzon, John Jones, Mark Dennis
February 17, 2013

www.pace-monmouth.org

PROGRAM FOR ACCELERATION IN CAREERS OF ENGINEERING

Object Oriented O-O Concepts in Programming

Forest is an area with a high density of **trees**

Class →

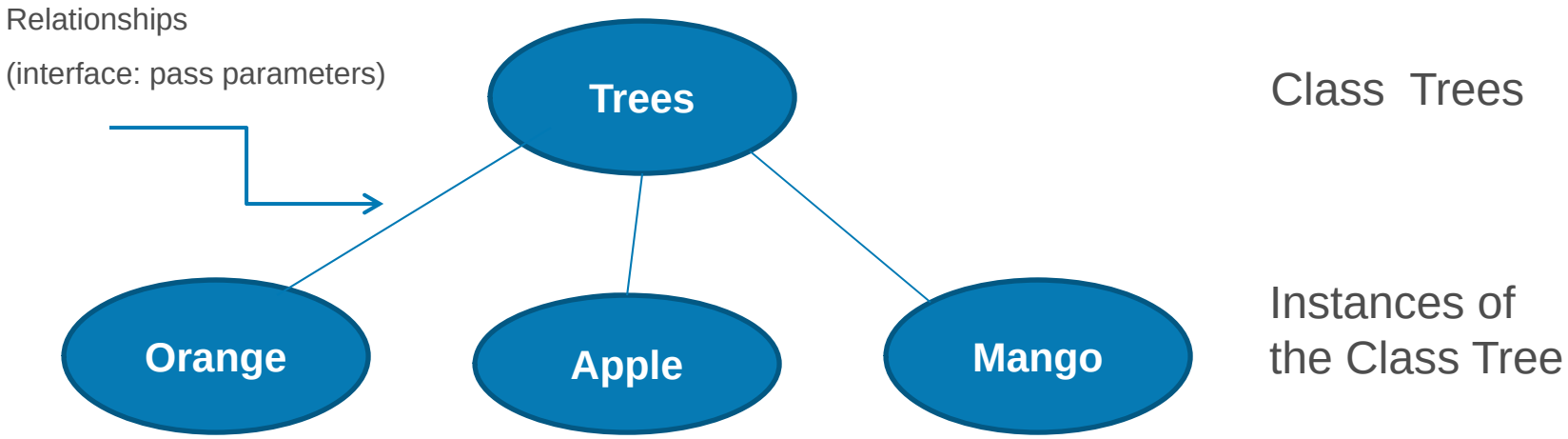
Instances of the Class ↓

Orange Tree

Apple Tree

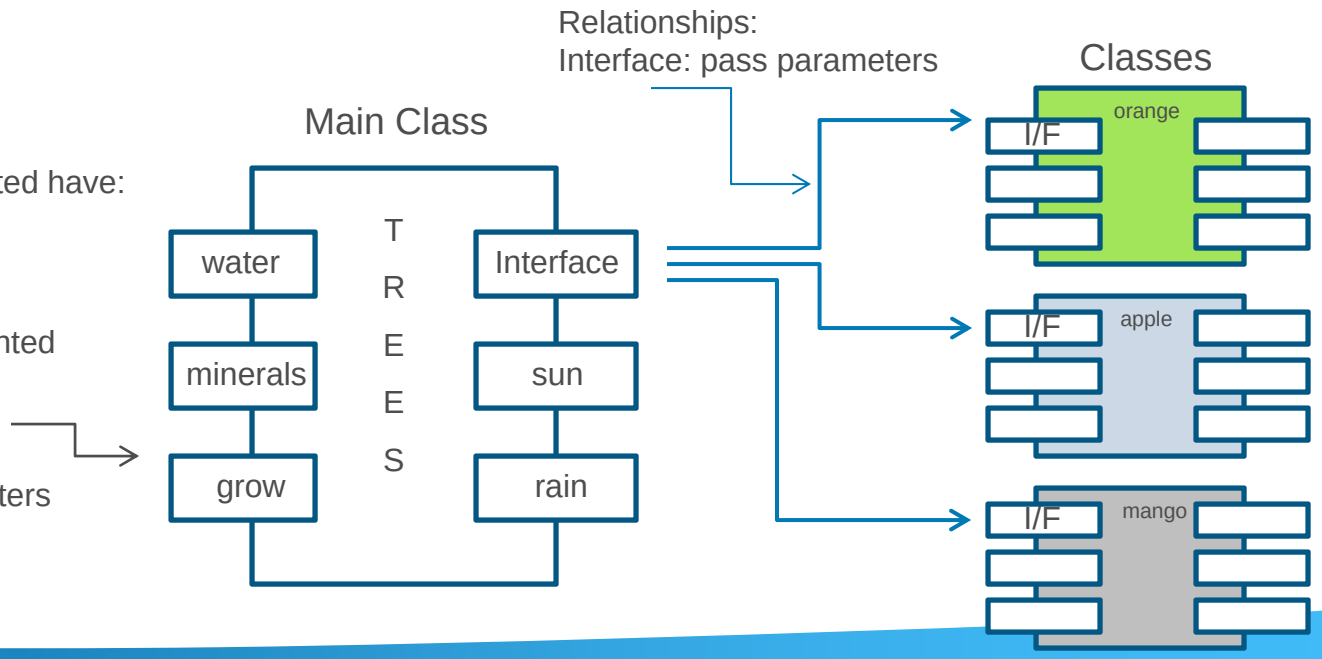
Mango Tree

Example 1: a Class, Instance of a Class, Relationships

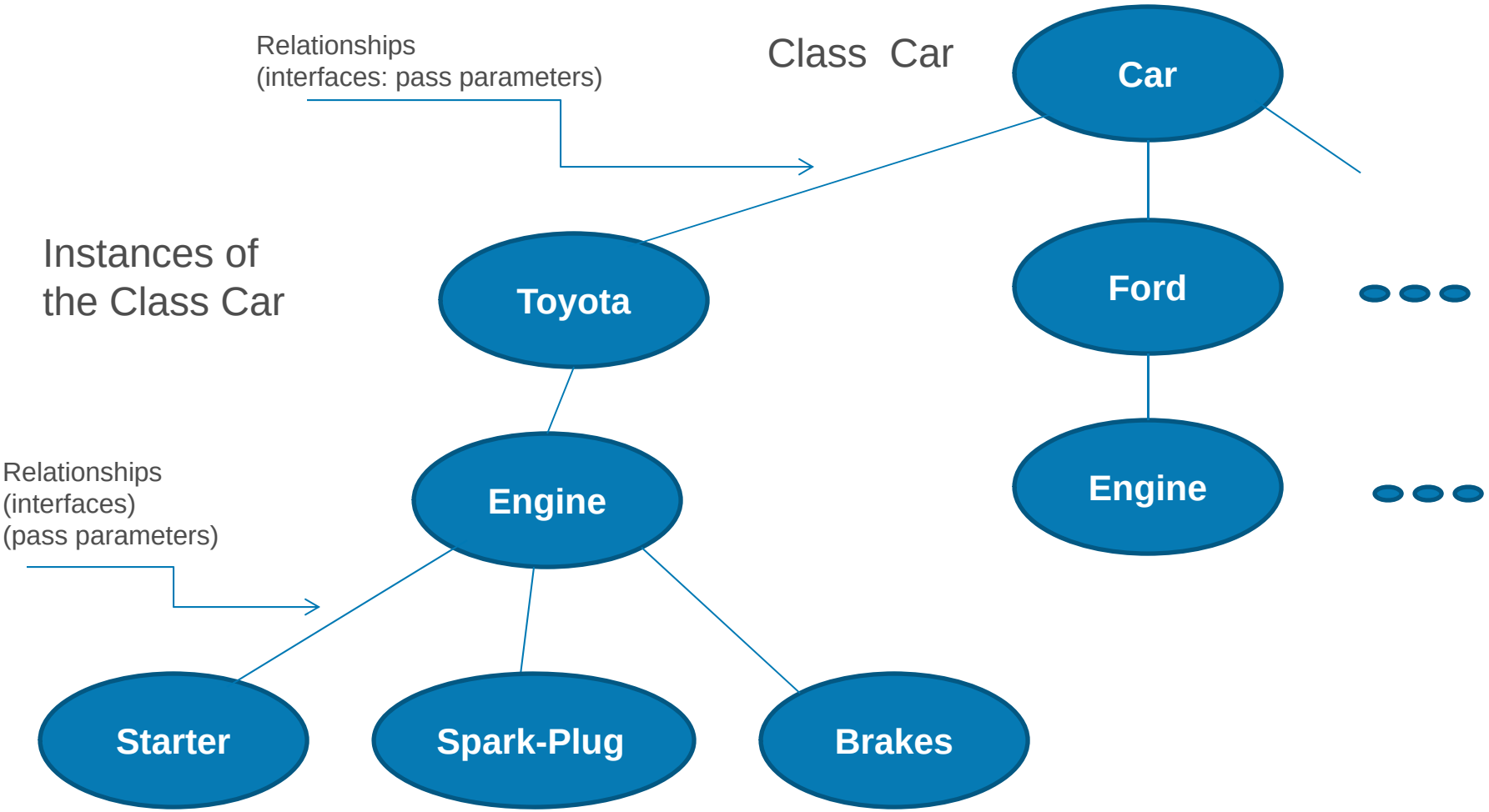


The Objects in Object-Oriented have:
- properties and
- behaviors

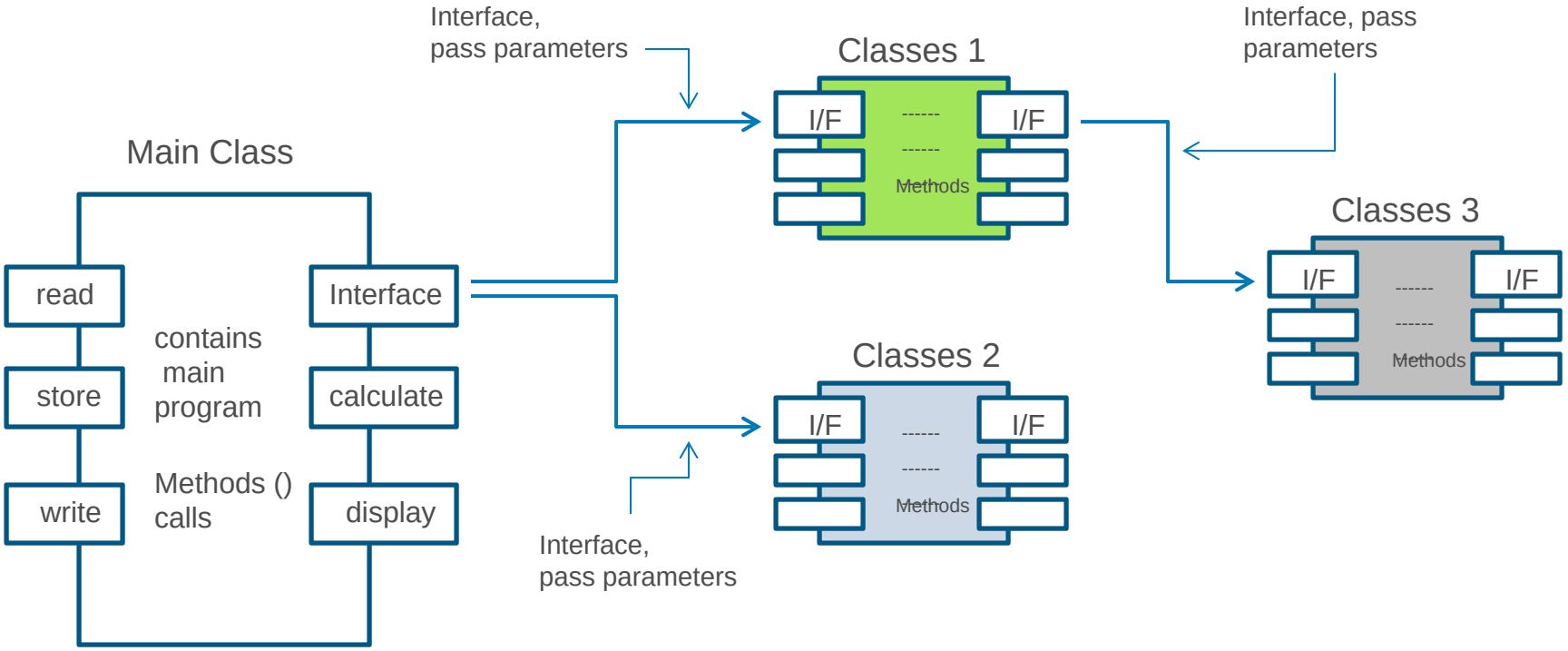
The Objects can be represented by classes, so they have:
- properties and
- behaviors, and
- Interfaces to pass parameters



Example 2: a Class, Instance of a Class, Relationships



Example 3: a Class, Instance of a Class, Relationships



Main Class

main program ( )

call method1 in class1

call method1 in class2

Class1

call method1 in class3

call method2 in class3

Importing a program from the "Java Program Library" Class

- There are times when we are allowed to use Java programs already written for you to use; for example, the Scanner program.
- Scanner allows you to "interact" with the user via the Keyboard.
- Scanner is located in the "Java Program Library". Call it, and use it.

```
import java.util.Scanner;  // get the Scanner program from the Java Library
```

- Once you call Scanner, you need to create an object to reserve memory for use.

```
Scanner input = new Scanner (System.in);
```

A Class and a Main Program in Java

A Java program needs a Main Class, and a main program to run, e.g., main ( ).

For example:

public class name and

public static void main (String [ ] args).

Creating a Simple Class:

A sample of a class with its simple form, a class that contains only variables. In essence, such a class defines a "compound of data type" that consists of its individual data number. Consider the following listing of variables:

```
class Sample {   // BEGIN
    int a;
    float b;
      public static void main (String [] args) {  // Begin
          a = 20;  b = 20.56f;
          System.out.println (" $ " + a + "      $ " + b);
       }    // End of main program
}    // End of Class
```

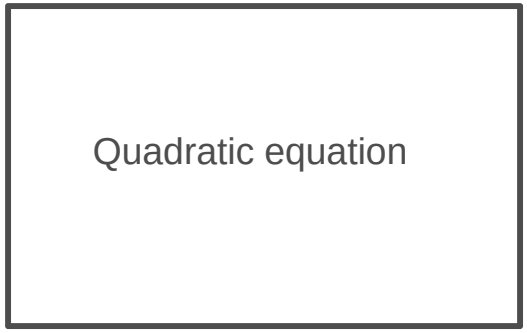
This declaration defines a **class** called Sample that consists of one **integer** number, a, and one floating point number, b. This "declaration" does not actually create any **objects.** It simply defines the form of an object. An object is created via the **new** operator.

Object (class) encapsulation; hiding information, example

Remember when you wrote the **while Loop** for the Quadratic Equation (ax^2 + bx + c = 0) so we can repeat all the calculations until the coefficients were a = 0, b = 0 and c = 0?

What you did was to “encapsulate” the Quadratic Equation behavior, by hiding all the calculations.

while (condition) { // Begin



} // End of while loop

We can create a Main Class that invokes class 2 calculations by passing the a, b, c, root1, root2 parameters.
Class 2, Quadratic Equation, calculates the roots and passes those parameters to the Main Class.
The Main Class receives root1 and root2 parameters and display them to the user.