

Computer Programming Methodologies

Sequential Programming

- lines of code are written and executed one after the other
- disadvantage - sometimes have to repeat the same code over and over again.

Procedural Programming

- lines of code are grouped together into a bundle called a **procedure** (in *Java*, we call them **methods**) to perform a certain task.
- efficient – reuse the same code, no need to write same code over and over
- shorter program

Object-Oriented Programming (OOP)

- separate file called **classes** (remember Scanner?)
- classes define data / data structure
- classes have their own methods that can be applied to their data
- classes can interact with each other to create a working program

Advantages of OOP

- code is no longer in one big file
- classes are separate files that can be worked on by different people at the same time
- the classes are reusable (for example libraries, or API's)

Native Classes

The following table lists some of *Java*'s **native classes**.

<i>String</i>	textual data (e.g., a string of characters)
<i>Boolean</i>	logical states (e.g., true and false)
<i>double</i>	floating-point numbers (e.g., numbers with a decimal value)
<i>int</i>	integer numbers (e.g., no decimal value)
<i>Array</i>	an ordered list
<i>Date</i>	a specific point in time
<i>Math</i>	common mathematical values and operations

<pre>int x1 = 3;</pre>	<p>The class is "int" or integer and the object or instance the integer class is "x1". The data or property(s) of this class is a whole number.</p>
<pre>String s1 = "sample"; String s2 = "another example";</pre>	<p>s1, and s2 are instances of the class String. s1 and s2 are two distinct objects even though they are from the same class. "s1" and "s2" are instance variables.</p>

JAVA CLASSES

A class is a blueprint or a template for creating different objects which defines its properties and behaviors. Java class objects exhibit the properties and behaviors defined by its class. A class can contain fields and methods to describe the behavior of an object.

Custom classes can be written from scratch to produce specialized types of content such as a car in a racing program. *Java* also provides pre-made classes (native classes) that are used to manipulate basic types of information, such as numbers and text.

Example:

An example of a class could be two integers. Objects (variables) could be created from this class. Every object (variable) created from this class would have two integers (because this is what the blueprint states for this class).

Activity

A class is a collection of data for example two integers could be a class. An object is a variable that is created with the data structure

1. Create a class/object using the block pieces – let one colour (or size) represent one primitive data type (for example integer).
2. Create any combination you would like and join the pieces together to create an object: 2 integers, 1 integer + 1 double, 1 double + 1 string + boolean, the combinations are limitless
3. identify the data types & corresponding colours

<pre>public class My2DPoint { double x,y; }</pre>	<p>The class name is "My2DPoint". The data or property(s) of this class is: two numbers (of type double).</p> <p>Convention: all classes should start with a Capital letter</p>
---	---

<pre>public class My2DPoint { double x,y; // constructor public My2DPoint () { x = 3; y = 5; } }</pre>	<p>Constructor – must be named exactly as the class name, a special method that is invoked when a new object is created using the "new" key word such as:</p> <pre>new My2dPoint();</pre>
---	---

<pre>public class TestProgram { public static void main (String [] args) { int x1 = 0; My2DPoint p1; p1 = new My2DPoint(); System.out.println (" p1.x = " + p1.x); System.out.println (" p1.y = " + p1.y); My2DPoint p2 = new My2DPoint (); System.out.println (" p2.x = " + p2.x); System.out.println (" p2.y = " + p2.y); p2.x = 25.5; p2.y = 12.54; System.out.println (" p2.x = " + p2.x); System.out.println (" p2.y = " + p2.y); } }</pre>	<p>Constructor is invoked and creates a new object called "p1"</p> <p>2nd object this time called "p2"</p> <p>p2 data being assigned new values</p>
--	---

Exercise:

Create a class called "My3dPoint" that contains 3 numbers; x, y, z (double).

Create a "Test" class that creates two objects; "point1" & "point2" using the "My3dPoint" class.

Display the properties of "point1" & "point2".

Change the values of the 3 numbers in point1 & in point2.

Display the new properties of "point1" & "point2".

Home Work.

Create a class called "Student" that contains name, student number, and grade.

Create a "Test" class that creates two objects using the "My3dPoint" class.

Display the properties of the two objects.

Change the properties of the 2 objects.

Display the new properties the two objects.