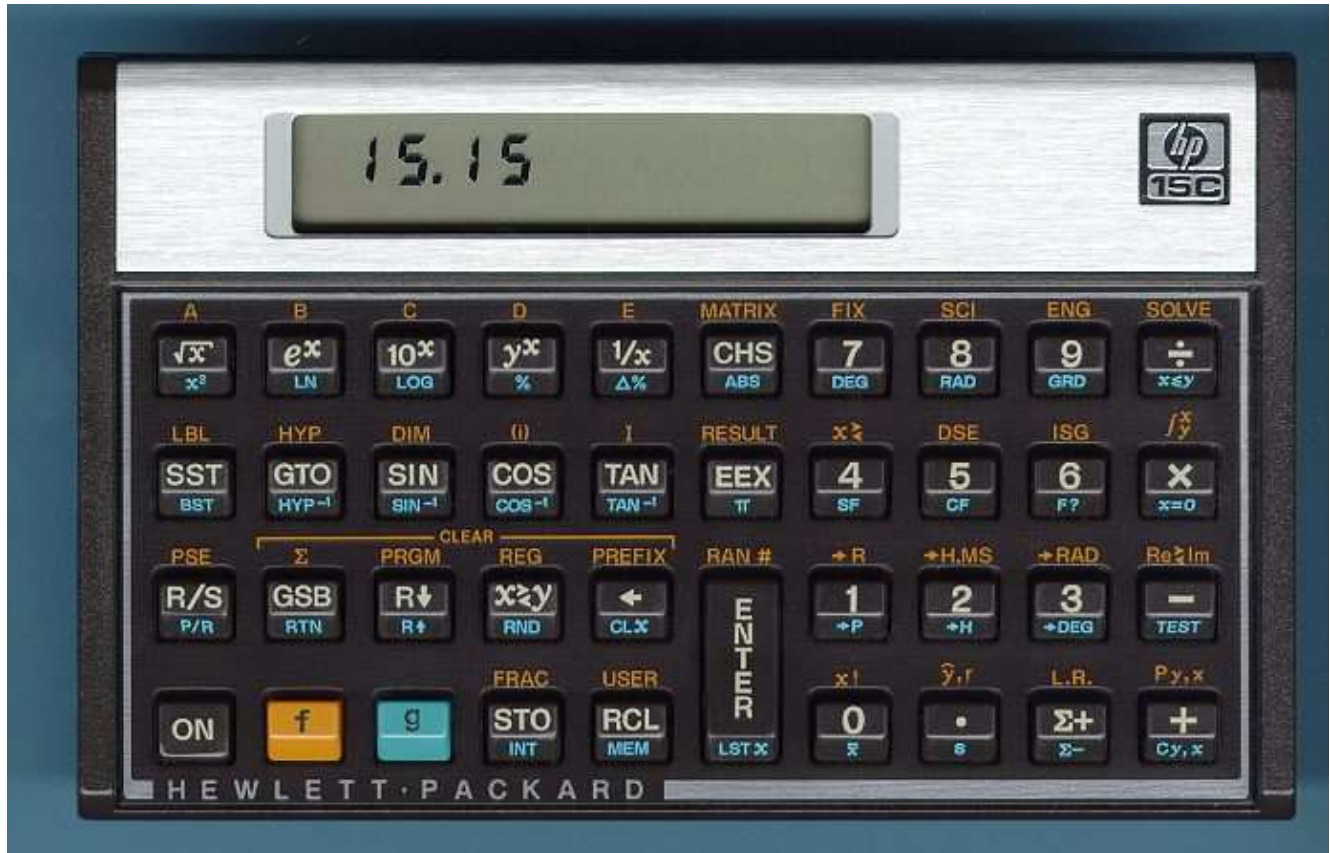


Programming Assignment 3: Stack Calculator

Based on Chapter 6, programming project 1

Description:

Hewlett-Packard has a tradition of creating stack-based calculators. Rather than using standard algebraic notation ($1 + 1 =$), the user would enter the values, then the operator. The calculator had an “enter” key to push each value onto a stack, then pop the stack when an operation key (e.g. “+” or “-”) was pressed.



Assignment:

Create a Calculator class that implements the provided StackCalculator interface

Feature	Signature	Requirement
Constructor	Calculator()	Precondition: <i>none</i> Postcondition: a new calculator with an empty number stack
Methods	public void enter(String entry)	Precondition: entry is either a double value

		<p>a recognized operator [+, -, *, /]</p> <p>Postcondition: if the entry is a double, the value is pushed onto the calculator's number stack. If an operator, the top two values are popped from the stack, the operation performed, and the result pushed onto the stack. Note: the second operator is popped first.</p>
	double peek()	<p>Precondition: number stack is not empty, otherwise throws EmptyStackException</p> <p>Looks at the number at the top of this stack without removing it from the stack.</p>
	double pop()	<p>Precondition: number stack is not empty, otherwise throws EmptyStackException</p> <p>Removes the number at the top of this stack and returns that value.</p>
	void clear()	Removes all numbers from this stack. The stack will be empty after this call returns
	boolean isEmpty	Tests If the number stack of this calculator is empty
	int size()	Returns the number of values in this calculators number stack.

Specifications:

	Requirements
Project Submission	<p>Your submission shall be an exported Eclipse project zip file</p> <p>The name of the project shall be in the form:</p> <p><i>P3_AAAnnn_Calculator</i></p> <p>AAAnnn will serve as your “student identifier”, where</p> <p>AAA – your initials (two or three characters)</p> <p>nnn – the last 3 digits of your Metro State student ID</p> <p>The jar file must have the Java sources</p> <p>Your classes must be in a package named with the following prefix:</p> <p style="padding-left: 40px;">edu.metrostate.ics240.p3.aaannn.calc</p> <p style="padding-left: 40px;">where <i>aaannn</i> is the student identifier as described above</p>

Code	<p>The Constructors and methods defined in the capabilities table will be public, spelled as specified, and with the return type as specified (void otherwise)</p> <p>Your submissions will be tested with an automated testing framework</p> <p>You must also provide a test class, separate from the calculator, showing your test cases</p> <p>it does not have to be a JUnit test, but that would be highly recommended.</p> <p>Your tests should demonstrate how you ensure your work meets the specification.</p> <p>The test should validate preconditions and boundary cases</p> <p>Your code must be free of compile-time errors</p> <p>Your code must comply with Java coding conventions</p>
------	---