**Program Fifteen Version one: (uses the command-line argument)**

//summary: This program uses postfix notation to caculate an equation. it takes in args (which would

//have been set to postfix notation) and solves/outputs the quation.

//name: Jenna Wolf

//class: Fundamentals of Programming, CS155 - 01

//instructor: Dr. Art Kazmierczak

//date: 11/27/2023

import java.util.Stack; //allows the stack class to be used

public class Main

{

public static void main(String[] args)

{

Stack<Double> math = new Stack<>(); //holds the math data

String[] expression = args; //holds the expression data and is set to args

//for loop that iterates until the length of expression is hit

for(int i = 0; i < expression.length; i++)

{

//goes as long as string can be made into double

try

{ //sets temp to a double and adds it to the top of math stack

double temp = Double.parseDouble(expression[i]);

math.push(temp);

}

catch(NumberFormatException e) { //catches operator

}

//checks to see if we are at an operator

if(expression[i].charAt(0) == '+' || expression[i].charAt(0) == '-' ||

expression[i].charAt(0) == '\*' || expression[i].charAt(0) == '/')

evaluate(math, expression[i].charAt(0)); //calls the evalute method

}

System.out.println("Answer: " + math.pop()); //outputs the answer

}

public static void evaluate(Stack<Double> math, char op)

{

double num1 = math.pop(); //holds the num1 data and sets it to the top of math

double num2 = math.pop(); //holds the num2 data and sets it to the top of math

if(op == '+') //adds the numbers

math.push(num1 + num2);

else if(op == '-') //subtracts the numbers

math.push(num1 - num2);

else if(op == '\*') //multiplies the numbers

math.push(num1 \* num2);

else //divides the numbers

math.push(num1 / num2);

}

}

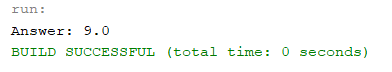
**Version two: (tests out the postfix notation. Used for output)**

Stack<Double> math = new Stack<>(); //holds the math data

String equation = "1 2 + 3 \*"; //holds the equation data and sets it

String[] expression = equation.split(" "); //holds the expression data and is set to the split of equation

**Output:**

****