

# **Software Requirements Specification**

EECS 2311—Software Development Project

19 February 2016

# Postfix Calculator

## Software Requirements Specification

1.1.0

February 19, 2016

Team 3

Prepared for  
EECS 2311—Software Development Project  
Instructor: **Gunnar Gotshalks**  
Winter 2016

## Revision History

Date	Description	Author	Comments
02/13/16	Version 1.1.0	Brooks Dulla	Base version

## Document Approval

The following Software Requirements Specification has been accepted and approved by the following group members:

Signature	Printed Name	Role	Date
	Andrew Ferreira	Testing & Documentation	
	Brooks Dulla	Documentation	
	Dan Sheng	Documentation	
	Yari Yousefian	Testing & Development	

# Table of Contents

<b>REVISION HISTORY .....</b>	<b>II</b>
<b>DOCUMENT APPROVAL.....</b>	<b>II</b>
<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 PURPOSE.....	1
1.2 SCOPE .....	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS .....	1
1.4 REFERENCES.....	1
1.5 OVERVIEW.....	1
<b>2. GENERAL DESCRIPTION.....</b>	<b>1</b>
2.1 PRODUCT PERSPECTIVE.....	1
2.2 PRODUCT FUNCTIONS .....	1
2.3 USER CHARACTERISTICS .....	2
2.4 GENERAL CONSTRAINTS .....	2
2.5 ASSUMPTIONS AND DEPENDENCIES .....	2
<b>3. SPECIFIC REQUIREMENTS.....</b>	<b>2</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS .....	2
3.1.1 <i>User Interfaces</i> .....	2
3.1.2 <i>Hardware Interfaces</i> .....	<i>Error! Bookmark not defined.</i>
3.1.3 <i>Software Interfaces</i> .....	<i>Error! Bookmark not defined.</i>
3.2 FUNCTIONAL REQUIREMENTS .....	2
3.2.2 <i>&lt;Functional Requirement or Feature #2&gt;</i> .....	3
3.3 USE CASES .....	3
3.3.1 <i>Use Case 1</i> .....	3
3.3.2 <i>Use Case 2</i> .....	3
3.3.3 <i>Use Case 3</i> .....	3
3.3.4 <i>Use Case 4</i> .....	3
<b>4. NON-FUNCTIONAL REQUIREMENTS .....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
REQ 1.....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
REQ 2.....	<b>ERROR! BOOKMARK NOT DEFINED.</b>

## 1. Introduction

### 1.1 Purpose

This document outlines the requirements the end-product *Postfix Calculator* should meet, with special acknowledgement to the intended user-base and the scenarios in which the program is intended to be used in.

### 1.2 Scope

*Postfix Calculator* program is intended to be an equivalent to the handheld calculator, allowing users to calculate the result of applying operations on input values. It should have an interface that emulates a standard calculator with a similar set of functions and operations, namely: addition, subtraction, multiplication, division, factorial, negation and basic trigonometric functions. The calculator will also implement a number of other features, including: enter, undo, clear, graphing view, as well as range and value selection to enable more complex operations. The *Postfix Calculator* will deviate from traditional calculators in the way values and operations are entered into the program; values are to be listed separately from the operators that are to be applied.

### 1.3 Definitions, Acronyms, and Abbreviations

rad

### 1.4 References

-Any resources referenced

### 1.5 Overview

*Still need to add:*

- Describe the remaining sections that are going to be addressed in the document

## 2. General Description

### 2.1 Product Perspective

The product is designed to run on any Operating System capable of executing a "Runnable JAR" file (*list which operating systems and cite*). This includes Unix and Windows Operating systems. The only requirement is installation of the Java Runtime Environment on the host machine.

### 2.2 Product Functions

The *Postfix Calculator* program will receive user input from a button pad. The input may consist of digits, functions or operations. The program interface will allow access to the following

operations: addition (+), subtraction (-), multiplication (x), division (/), factorial (!), negation ((-)), sin function (sin) and cos function (cos). In addition to these operators and functions the program will enable the following features: clear, undo, select, value, enter, rad, pi, graphing view and decimal point.

### 2.3 User Characteristics

An individual interested in performing mathematical operations that has access to a machine with a point and click interface capable of running the executable *Postfix Calculator* program.

### 2.4 General Constraints

The *Postfix Calculator* will work on any system with the Java Runtime Environment installed, which excludes some mobile devices (**WHICH**). Initial prototypes may not support operator precedence rules supported by many traditional calculators.

### 2.5 Assumptions and Dependencies

Inputting values into the program in a postfix manner may not be intuitive or immediately apparent to users, as users are likely accustomed to inputting values in infix notation. Without a message detailing the specific operation of the calculator users may not use the program successfully.

## 3. Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

The program will operate as much as possible in the same manner that a handheld calculator operates. Starting the *Postfix Calculator* will open a window of fixed dimension for all screen sizes. The opened window will reveal interface with two displays: one for input of values and the corresponding outputs, and a second display for graphing purposes. As with a regular calculator the buttons are positioned below the displays and are grouped together in the following manner: digits 0-9, operands, functions and features. A single legible font of appropriate size is chosen for the entire interface. Appropriate error messages are displayed on the main display for unsupported input scenarios. Colors should contrast sufficiently to allow for legibility of fonts and different components of the interface. Displays and buttons should be immediately apparent, contrasting against their respective background panels.

### 3.2 Functional Requirements

#### 3.2.1 <Functional Requirement or Feature #1>

##### 3.2.1.1 Introduction

##### 3.2.1.2 Inputs

##### 3.2.1.3 Processing

##### 3.2.1.4 Outputs

##### 3.2.1.5 Error Handling

**3.2.2 <Functional Requirement or Feature #2>**

...

**3.3 Use Cases****3.3.1 Use Case 1**

Use Case ID:	UC-1		
Use Case Name:	Two-Operand Single Operator Expression		
Created By:	Brooks Dulla	Last Updated By:	Brooks Dulla
Date Created:	02/19/16	Date Last Updated:	02/23/16

Actor:	Student
Description:	Student enters two digits and performs addition
Preconditions:	Point-n-click enabled device; device executes runnable JAR files
Postconditions:	Result of operation displayed
Priority:	Essential - High
Frequency of Use:	Often
Normal Course of Events:	1.Student enters the number 5 2.Student enters the number 12 3. Student selects addition 4. Result is displayed
Alternative Courses 1:	1.Student enters the number 5 2.Student enters the number 11 3.Student chooses to undo the last entry 4. Student enters the number 12 5. Student selects addition 6. Result is displayed
Exceptions:	/
Includes:	/
Special Requirements:	/
Assumptions:	Student familiar with Postfix expressions
Notes and Issues:	/

**3.3.2 Use Case #2**

Use Case ID:	UC-2		
Use Case Name:	Multiple Operand Single Operator Expression		
Created By:	Brooks Dulla	Last Updated By:	Brooks Dulla
Date Created:	02/19/16	Date Last Updated:	02/23/16

Actor:	Student
Description:	Student enters three or more digits and performs division
Preconditions:	Point-n-click enabled device; device executes runnable JAR files
Postconditions:	Result of operation displayed
Priority:	
Frequency of Use:	
Normal Course of Events:	
Alternative Courses 1:	
Exceptions:	/
Includes:	/
Special Requirements:	/
Assumptions:	Student familiar with Postfix expressions
Notes and Issues:	/

### 3.5 Non-Functional Requirements

#### 3.5.1 Performance

#### 3.5.2 Reliability

#### 3.5.3 Availability

#### 3.5.4 Security

#### 3.5.5 Maintainability

#### 3.5.6 Portability

### 3.7 Design Constraints

*Design constraints imposed by some of the requirements outlined so far*

### 3.9 Other Requirements

*Any remaining requirements go here!!*