Scouts BSA Programming Project Guide

Benjamin Davis [benjamin.j.davis96@gmail.com]

Contents

Chapter 1

Introduction

In addition to the historical and general knowledge, the Scouts BSA Programming Merit Badge requires several programming capability demonstrations. Requirement 5 outlines provides this requirement as follows:

> **5.A** With your counselor's approval, choose a sample program. Then, as a minimum, modify the code or add a function or subprogram to it. Debug and demonstrate the modified program to your counselor.
>
> **5.B** With your counselor's approval, choose a second programming language and development environment, different from those used for requirement 5a and in a different industry from 5a. Then write, debug, and demonstrate a functioning program to your counselor, using that language and environment.
>
> **5.C** With your counselor's approval, choose a third programming language and development environment, different from those used for requirements 5a and 5b and in a different industry from 5a or 5b. Then write, debug, and demonstrate a functioning program to your counselor, using that language and environment.
>
> **5.D** Explain how the programs you wrote for requirements 5a, 5b, and 5c process inputs, how they make decisions based on those inputs, and how they provide outputs based on the decision making

These requirements essentially boil down to 3 requirements in 3 different programming languages and the ability to explain them. While you are certainly able to choose any program of your desire (within the limits of your counselor's stipulations) this can be quite overwhelming and challenging for anyone without pre-existing knowledge of programming and a fair bit of programming experience. Thus I have found most scouts are more successful when they are provided with a pre-defined and recommended set of programs to use as a starting point. This guide is intended to provide all of the background information and instructions necessary for a scouts success in this portion of the merit badge.

1.1 Getting Started

The first thing we need to start programming is a computer. Modern computers come in a wide variety from industrial servers to simple laptops to gaming computers. The projects presented here are relatively simple and should be capable of operating on virtually any class of computer. That said there is a difference

between computers which will impact the development process: operating system. I cannot recommend Linux highly enough if you desire to do any substantial amount of programming; however it is still the minority in computer operating systems. Thus I have verified each of the projects here to operate on Windows 10, Windows 11, Ubuntu, and Debian (Currently I do not have access to a Mac to verify these projects on that operating system, but I have no reason to believe Mac will not work with these projects). A deeper discussion of developing in these OS's can be found in Appendix A.

In order keep all of the files necessary for this merit badge in a single controlled space I have elected to use GitHub to host the files.

1.2 Projects

I highly encourage following the project requirements in order and using programs from the correct section for the respective requirement. While programs from `Project B` and `Project C` can be used interchangeably those found in `Project C` are notably more difficult, but also more rewarding. Below are the Projects which are recommended for each requirement. They have been placed in the order of difficulty that I expect them to pose to a brand new programmer. Additionally, they include the recommended program languages which could be used to most easily accomplish them (note some programming languages which we have discussed in the classroom portion are not found here, I am excluding them based on my experience that they tend to be overly complex without providing a novel advantage).

1.2.1 Project A Options

Web Calculator

Overview: Modify some code to complete a functioning calculator which operates out of a web-browser.

Programming Languages: JavaScript

1.2.2 Project B Options

Shape Area Calculator

Overview: Create a text-based program which can calculate the area of various shapes

Programming Languages: Python, Java, C++, C

Prime Number Finder

Overview: Create a text-based program which can calculate the first N primes

Programming Languages: Python, Java, C++, C, Fortran

Hang Man

Overview: Create a text-based HangMan game

Programming Languages: Python, Java, C++, C

1.2.3 Project C Options

If all of the below project options appear too difficult to accomplish in the given time or with your current school/extra-curricular activity schedule please talk to your counselor about using another option from the `Project B` list.

Sorting Numbers

Overview: Create a program which can sort a list of numbers using two different sorting algorithms.

Programming Languages: Python, C++, Java, C, Fortran

Self-Writing Program

Overview: Create a program can write its entire source code to a file

Programming Languages: C++, Java, C

Pong Game

Overview: Create a simple Pong-Style 2-D Game

Programming Languages: Python, BASIC, C++, Java

Snake Game

Overview: Create a simple Snake-Style 2-D Game

Programming Languages: Python, BASIC, C++, Java

Logic Puzzle

Overview: Create a simple program which provides the answer to a logic puzzle such as the Wolves and Sheep river crossing puzzle.

Programming Languages: Prolog, ALF, Python

FPGA Adder

Overview: Create device which can add two 8-bit numbers on an FPGA from switches and output the result to LEDs.

Programming Languages: SystemVerilog, Verilog, VHDL

Chapter 2

Project A

Chapter 3

Project B

Chapter 4

Project C

Appendix A

Developing in Windows vs Linux vs Mac

A.1 WSL: The Best of Both Worlds?

Appendix B

Language Installations