Program 3 Assignment: Bounty Hunter

→ Group Members: Khushbu Joshi, Abigail McCall

Description:

- This program will be a Bounty Hunter Simulation. You can enter people's bounties and their defining features. They can be searched and displayed by their name.

Classes:

- Binary tree
- Bounty

Number of files: 7

- BountyHunter.cpp
- Tree.h
- Tree.cpp
- Bounty.cpp
- Bounty.h
- makefile
- TESTCASE.txt

Functions:

- Constructors and Destructors
- Getters for variables (name, age, eye color, build, body markings, city)
- Display and search bounty functions located in tree.h
- Other functions for tree
    - Private helper functions and public methods

Program Flow:

- The program takes in bounties from user input. The user inputs bounty values, and then they can be displayed. It can also search bounties based on its name. The menu will allow the user to input these options or exit the program.

Initial Goal:

- The initial goal was to create a program that can accept the bounties and display them, demonstrating the use of trees.
    - This was successful, although some of the additional features we wanted to implement were sacrificed in the interest of time.

Looking back:

- Our process was much more productive this time than in the previous projects. We were able to coordinate better on what we needed to do and finish it sooner than later. Since by this point, we work better as a team. We were able to set up the files and work on each component individually.
- If we were to do something differently, I would be to communicate more on each aspect of the program. But other than that, there was not much that needed changing.
- We learned how to use trees as a data storage class and implement them as a program.

Outcome:

- The program should be able to read the values and display them properly. The files have been tested through each path and are included in the TESTCASE.txt file.
- The program uses a makefile so that should be taken into account for testing.