|  |  |
| --- | --- |
| **Application/ Program name:** | FiveDice2&FiveDice3 |
| **Written by:** | Zachary Muerle |

|  |
| --- |
| **Purpose or problem definition:** |
| In Chapter 4, you created a Die application that randomly “throws” five dice for the computer and five dice for the player. The application displays the values. Modify the application to decide the winner based on the following hierarchy of Die values. Any higher combination beats a lower one; for example, five of a kind beats four of a kind. when both players have the same combination of dice, the higher value wins. For example, two 6s beats two 5s. |
|  |
| **Program Procedures:** |
| Create 2 arrays of 5 dice, roll them all, find the highest n of a kind the array has, and the value of that n of a kind. Compare the 2 values: if the n is the same, the higher value wins. if both are equal, everyone loses completely. |
|  |
| **Algorithm/Processing/Conditions:** |
| **Inputs: none** |
|  |
| **Processes: main,** getScoreName, containsNOfAKind |
| **(see source code- it has comments. Most of these do exactly what they sound like)** |
| **Outputs: info** |
| The player’s and computer’s dice, and the winner, in detail |
|  |
| **Notes & Restriction:** |
| The program is a bit snarky when there’s a tie. |
|  |
| **Comments:** |
| This was surprisingly difficult, given the prompt, it seemed really easy, took a bit longer than expected, but not much. |