Ch 6 HW Example — DeathSpank Black-Market Horse Races

Discrete Structures: Permutations, Combinations, Stars&Bars, and Probability

(a cheeky caper in the *Thongs of Virtue* universe)

Student: _____

Partner: _____

November 12, 2025

Contents

This writeup follows the Universe/Counting/Python/Probability/Reflection structure from the Chapter project brief. :contentReference[oaicite:0]index=0

How to Read This

This document is the main entry point (`Ch_6_HW_Example.tex`). Each chapter lives under `chapters/`. Build with `make`. We keep the title verbose so future-us never opens the wrong PDF at 3 AM.

Chapter 1

Core Story: Black-Market Horse Racing in DeathSpank's World

In the smoky back-alleys of Spanktopia, the *Bureau of Questionable Athletics* runs unsanctioned races featuring dubious "horses": skeleton steeds, thonged unicorns, and a mysterious two-coconut contraption named *Sir Clip-Clop*. Bettors whisper in code, merchants haggle over ethically ambiguous oats, and the odds are set by an enchanted abacus with a drinking problem.

1.1 Playable Loop (Student-friendly summary)

Step 1: Choose a race card: a set of n entrants from a stable of N creatures.

Step 2: Assign lanes and start order: who lines up where (order matters).

Step 3: Distribute upgrade points: illegal mods (glitter horseshoes, turbo thongs) across k stats.

Step 4: Place a bet: pick winners, exactas, or "chaos trifecta."

Step 5: Run the race: outcomes are sampled; payouts depend on combinatorics-based odds.

1.2 Entities

- **Stable:** N entrants. Example archetypes: Bone Pony, Thongicorn, Greased Donkey, Sir Clip-Clop.
- **Race Card Size:** n selected to race.
- **Stats:** Speed, Skullduggery, Stamina, Sparkle ($k = 4$ stats).
- **Illegal Mod Points:** p points distributed across the k stats (with or without caps).

1.3 Constraints (examples we'll count later)

- Exactly n of N entrants race (order does not matter for selection).
- Lanes are a permutation of the chosen n (order matters).
- Mod points: stars-and-bars across k stats, optionally with per-stat caps.

Chapter 2

Counting Toolkit: Where the Math Meets the Mayhem

2.1 Combination: Choosing the Race Card

Problem: From N creatures, pick n to race (order does not matter).
Model: $\binom{N}{n}$ (combination).
Example: If $N = 10$ and $n = 4$, then $\binom{10}{4} = 210$.

2.2 Permutation: Assigning Start Order/Lanes

Problem: Arrange the n chosen racers on the track.
Model: $n!$ (permutation of n distinct entrants).
Example: For $n = 4$, there are $4! = 24$ lane orders.

2.3 Stars and Bars: Distributing Mod Points

Problem: Distribute p identical mod points among k stats.
Uncapped Model: Number of solutions to $x_1 + \cdots + x_k = p$ in nonnegatives is $\binom{p+k-1}{k-1}$.
Example: $p = 6$, $k = 4 \Rightarrow \binom{9}{3} = 84$.
With per-stat cap c: Use inclusion–exclusion or casework (we'll demo both).

2.4 Probability From Counts

Template: $\mathsf{Prob}(\text{event}) = \dfrac{\#\{\text{favorable}\}}{\#\{\text{all equally likely}\}}$.
We will compute event sizes with the tools above, then verify via Python.

Chapter 3

Python Experiments: Formula Checks and Simulation

3.1 What We'll Test

- Verify $\binom{N}{n}$ by enumerating all n-subsets for small N.
- Verify $n!$ lane orders by enumerating permutations for small n.
- Verify stars-and-bars counts via generating all integer k-tuples summing to p.
- Estimate betting event probabilities by Monte Carlo and compare to theory.

3.2 Sample Snippets (will live in `scripts/`)

Combinations and Permutations Check

```
from itertools import combinations, permutations
from math import comb, factorial

N, n = 6, 3
all_teams = list(combinations(range(N), n))
assert len(all_teams) == comb(N, n)

all_orders = list(permutations(range(n)))
assert len(all_orders) == factorial(n)
print("OK: comb/permutation counts match theory.")
```

Stars-and-Bars Enumeration (uncapped)

```
def stars_and_bars(p, k):
    # yield all k-tuples of nonnegatives summing to p (small p,k only)
    if k == 1:
```

```
        yield (p,)
        return
    for x in range(p+1):
        for rest in stars_and_bars(p-x, k-1):
            yield (x,) + rest

p, k = 5, 4
tuples = list(stars_and_bars(p, k))
from math import comb
print(len(tuples), "vs", comb(p+k-1, k-1))
```

Monte Carlo Outline

```
# Pseudo-code for a betting event (e.g., racer A finishes first):
# 1) Randomly generate lane order uniformly from permutations.
# 2) Apply a simple performance model (e.g., score = Speed + noise).
# 3) Sort by score to get finish order; record event.
# 4) Repeat many times; compare frequency to theoretical estimate.
```

Chapter 4

Probability: From Counts to Odds (with Flavor)

4.1 Exacta (Pick 1st and 2nd in Order)

All outcomes: $n!$.
Favorable (specific pair in exact order): Fix 1st, fix 2nd, permute rest $(n-2)!$.

$$\mathsf{Prob}(\text{Exacta on (A,B)}) = \frac{(n-2)!}{n!} = \frac{1}{n(n-1)}.$$

4.2 Trifecta (Pick 1st, 2nd, 3rd in Order)

$$\mathsf{Prob}(\text{Trifecta (A,B,C)}) = \frac{(n-3)!}{n!} = \frac{1}{n(n-1)(n-2)}.$$

4.3 Build With Mods: Capped Stars&Bars Example

Let $k = 4$ stats, p points, per-stat cap c. Number of assignments:

$$\sum_{j=0}^{\lfloor p/(c+1) \rfloor} (-1)^j \binom{k}{j} \binom{p - j(c+1) + k - 1}{k-1}.$$

We'll confirm small cases in Python.

Chapter 5

Reflection: What the Enchanted Abacus Taught Us

- What surprised you about universe size (e.g., how fast $n!$ explodes)?
- Where did modeling get messy (caps, constraints, "illegal" combos)?
- Do Monte Carlo results align with theory for small n? Why/why not?
- How could simple AI/search help balance racers or discover strong builds?

Appendix A

Appendix: Minimal Play Rules

A.1 Setup

Choose n racers from N; distribute p mod points across $k = 4$ stats (caps optional).

A.2 Race Resolution (toy model)

Each racer gets a performance score:

$$\text{Score} = \alpha \cdot \text{Speed} + \beta \cdot \text{Skullduggery} + \gamma \cdot \text{Stamina} + \delta \cdot \text{Sparkle} + \text{Noise},$$

rank by score to produce the finish order. Bettors win based on exacta/trifecta rules above.

A.3 Parameter Suggestions

Small demo: $N = 6$, $n = 4$, $p = 6$, $k = 4$, caps $c = 4$, noise $\sim \mathcal{N}(0, 1)$.