1. Default Seed (Same Sequence Every Run)

```python
# Without seeding.
import random

for k in range(1, 6):
    random_decimal = random.random()
    print(f"Random decimal {k}: {random_decimal}")
```

2. 7 Decimal Places

```python
# Generate random decimals with exactly 7 digits after the decimal.
import random

for k in range(1, 6):
    random_decimal = random.random()
    # Format to 7 decimal places using string formatting
    print(f"Random decimal {k}: {random_decimal:.7f}")
```

3. Fixed Seed (Consistent Output)

```python
# Using a fixed seed (42) ensures the same sequence every run.
# Why 42 ? It is Answer to the Ultimate Question of Life, Universe, and Everything
import random

random.seed(42)  # Explicit fixed seed

for k in range(1, 6):
    random_decimal = random.random()
    print(f"Random decimal {k}: {random_decimal:.7f}")
```

4. Dynamic Seed (Time-Based)

```python
# Using the current time as a seed produces different sequences each run.
import random
import time

random.seed(int(time.time()))  # Seed with current time (dynamic)
```

```
for k in range(1, 6):
    random_decimal = random.random()
    print(f"Random decimal {k}: {random_decimal:.7f}")
```

5. Coin Toss Simulation

```
# Simulate a coin toss with formatted output (H for heads, T for tails).
import random
import time

random.seed(int(time.time()))  # Seed with current time

# Print formatted headers
print(f"{'Toss #':<10}{'Result':<10}")
print("----------------------")

for k in range(1, 11):
    rand_num = random.random()
    result = 'H' if rand_num <= 0.5 else 'T'
    print(f"{k:<10}{result:<10}")  # Left-aligned columns with width 10
```

6. Dice Roll Simulation

```
# Simulate a dice roll using fixed seed
import random

random.seed(42)  # Set fixed seed for predictable results

for k in range(1, 6):
    die1 = random.randint(1, 6)  # Generate die roll (1-6)
    die2 = random.randint(1, 6)  # Generate second die roll
    total = die1 + die2

    # Format output to match C++ version's structure
    print(f"Toss {k} => Die 1: {die1}, Die 2: {die2}, Total: {total}")
```

7. Dice Roll Simulation(User Controlled Seeding)

```
# Simulate a dice roll using user's seed
import random

# Get user inputs for dynamic seeding and toss count
seed_value = int(input("Enter random number seed: "))
```

```python
random.seed(seed_value)  # Set dynamic seed based on user input

num_tosses = int(input("How many times do you want to toss the dice? "))

# Simulate dice tosses based on user request
for i in range(1, num_tosses + 1):
    die1 = random.randint(1, 6)  # Generate first die (1-6)
    die2 = random.randint(1, 6)  # Generate second die
    total = die1 + die2

    # Format output to match original C++ structure
    print(f"Die 1: {die1}, Die 2: {die2} => Total: {total}")
```