EXTRA ASSIGNMENT: MonteCarlo Approximation

1. The goal is to calculate the expected value of a function $f(x)=x^2$ over a probability distribution $p(x)$. Here, $p(x)$ is the normal distribution $N(x;\mu=0,\sigma^2=1)$, which is a standard normal distribution with a mean of 0 and a variance of 1.

2. The expected value of the function $f(x)=x^2$ under the distribution $p(x)$ is denoted as $\bar{f}=E[x^2]$, and can be computed analytically using integration:

   $$\bar{f}=\int_{-\infty}^{\infty} N(x;\mu,\sigma^2)\, x^2\, dx$$

   For the standard normal distribution, the result of this integral is $\mu^2+\sigma^2$, which simplifies to 1 (since $\mu=0$ and $\sigma^2=1$)

3. Monte Carlo is used as a method to estimate the expected value by sampling. Instead of performing the full integration analytically, we sample from the distribution and compute the mean of $f(x)=x^2$ over the samples. Its formula represents the average of the function values at N samples points x_n:

   $$\bar{f}\approx\frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

```python
# IMPORTED LIBS

import numpy as np
from numpy import random
import matplotlib.pyplot as plt

from ipywidgets import interact, interactive, fixed, interact_manual
import ipywidgets as widgets

import numpy as np
import matplotlib.pyplot as plt

# Define the function f(x) = x^2
def f(x):
    return x**2

# Total number of samples
N = 10000

# Generate N samples from a standard normal distribution (mean=0, 
standard deviation=1)
samples = np.random.normal(0, 1, N)

# Initialize list to store successive approximations
approximations = []
```

```python
# Calculate the Monte Carlo approximation successively
for i in range(1, N+1):
    # Average of the first 'i' samples
    current_mean = np.mean(f(samples[:i]))
    approximations.append(current_mean)

# Theoretical expected value E[x^2] = 1
true_value = 1

# Create the plot
plt.figure(figsize=(10, 6))
plt.plot(range(1, N+1), approximations, label="Monte Carlo
Approximation")
plt.axhline(y=true_value, color='r', linestyle='--', label="True Value
E[x^2] = 1")
plt.xlabel('Number of samples (N)')
plt.ylabel('Approximated E[f(x)]')
plt.title('Monte Carlo Approximation of E[x^2] with increasing N')
plt.legend()
plt.grid(True)
plt.show()
```