2. [**15 pt**] Solve

$$\boxed{y'' + K\,y = 0;\;\; y(1) = 0;\;\;\; y(5) + \Gamma\,y'(5) = 10}$$

using the nested-series-solution method shown in class.
For the series-solution answer, please work out the derivative of $y(x)$ as a series, too, so you can satisfy the boundary condition.

Here $K = \max([A, B, C])$ and $\Gamma = E + F + G$, where $A, B, C, D, E, F$ are taken from your TUID.

(a) Plot both your series solution and

(b) plot the exact solution on $1 \leq x \leq 5$ on the same graph for comparison.

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
TUID = [9, 1, 5, 1, 8, 7, 2, 8, 9]
LETTER_MAP = ['I', 'H', 'G', 'F', 'E', 'D', 'C', 'B', 'A']
total_sum = 0

for i in range(len(TUID)):
    total_sum += TUID[i]

average = total_sum / len(TUID)

print(f'my TUID average: {average}')

# Create a dictionary to map letters to integers
letter_to_int_map = {letter: integer for letter, integer in zip(LETTER_MAP, TUID)}

# Now, calculate the average of the letters of interest
K_letters = ["A", "B", "C"]
iota_letters = ["E", "F", "G"]

K = np.max([letter_to_int_map[letter] for letter in K_letters])
print(K)

iota = np.sum([letter_to_int_map[letter] for letter in iota_letters])
print(iota)
```

```
my TUID average: 5.555555555555555
9
14
```

## Nested Power Series Solutiuon

In [ ]:
```
# Number of terms for power series
N = 50

# Coefficient generation for power series
a = [0] * (N+1)
a[0] = 1
a[1] = 0

for p in range(N-1):
    a[p+2] = -9*a[p]/((p+2)*(p+1))
```

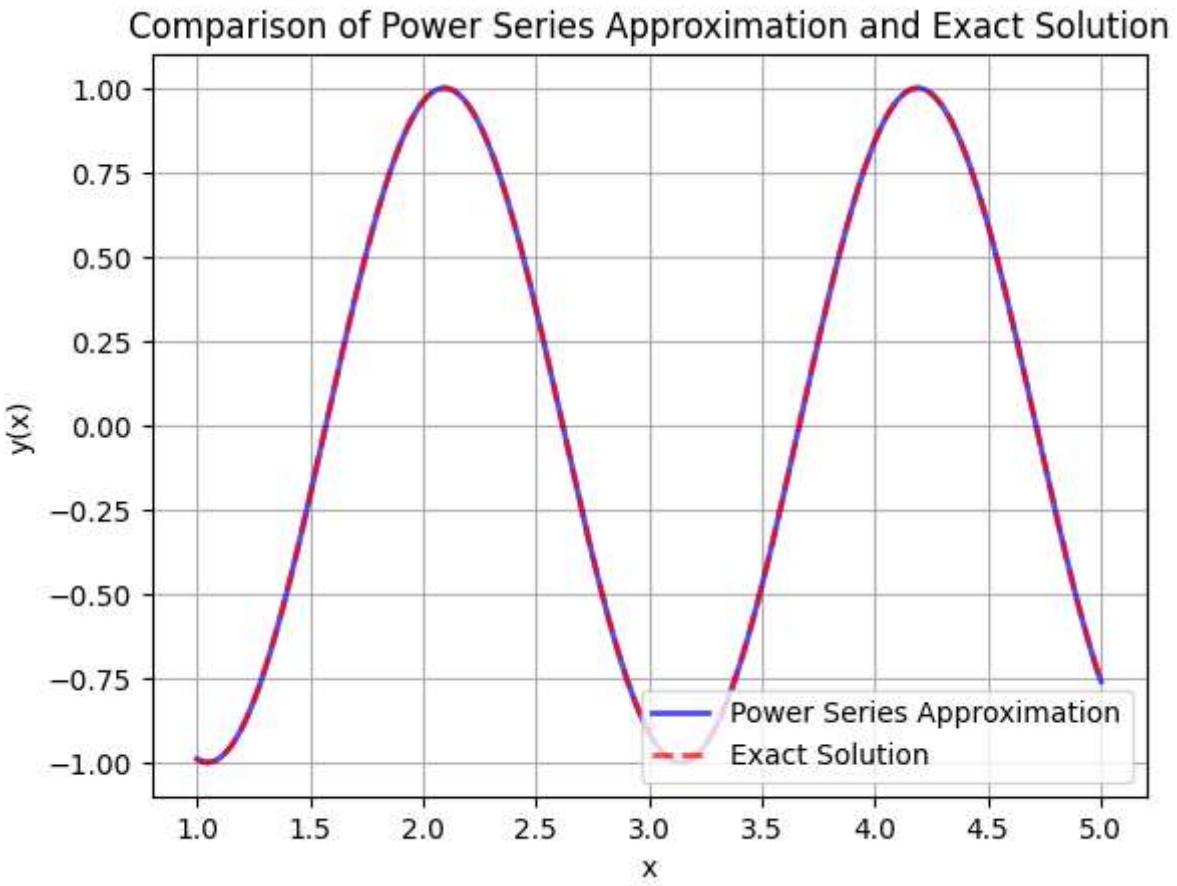
```
# Power series approximation
def y_approx(x):
    return sum([a[p] * (x**p) for p in range(N+1)])
```

## Exact Solution

In [ ]:
```
# Exact solution
C1 = 1
C2 = 0
def y_exact(x):
    return C1 * np.cos(3*x) + C2 * np.sin(3*x)

# Values for x
x_vals = np.linspace(1, 5, 400)
y_approx_vals = [y_approx(x) for x in x_vals]
y_exact_vals = [y_exact(x) for x in x_vals]
```

In [ ]:
```
# Plot
plt.plot(x_vals, y_approx_vals, label='Power Series Approximation', linewidth=2, co
plt.plot(x_vals, y_exact_vals, label='Exact Solution', linestyle='dashed', linewidt
plt.title('Comparison of Power Series Approximation and Exact Solution')
plt.xlabel('x')
plt.ylabel('y(x)')
plt.legend(loc='lower right')
plt.grid(True)
plt.show()
```


Comparison of Power Series Approximation and Exact Solution
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
y(x)
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
x
Power Series Approximation
Exact Solution