Experiment 1b:

Aim: To generate Hermite Polynomials using Rodrigues' formula and plot the first six Hermite polynomials .

Software packages used: language- Python3, packages- Numpy, Matplotlib, Sympy

Theory: The Rodrigues formula for Hermite Polynomials is given by:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2})$$

Code:

- Importing packages

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sympy import *
```

- Defining a function for Rodrigues formula

In [2]:
```
def hermite (n,y):
    x = symbols('x')
    output = (-1)**n* exp(x**2)* diff(exp(-x**2),x,n)
    return float(output.subs(x,y))

# this allows the defined function to take array input for y argument
hermite_polynomial = np.vectorize(hermite)
```

- generating 1000 points in the interval [-3,3] on the X-axis

In [3]:
```
X = np.linspace(-3,3,1000)
```

- generating Hermite polynomial y values for corresponding x values

In [4]:
```
H0 = hermite_polynomial(0,X)
H1 = hermite_polynomial(1,X)
H2 = hermite_polynomial(2,X)
H3 = hermite_polynomial(3,X)
H4 = hermite_polynomial(4,X)
H5 = hermite_polynomial(5,X)
```

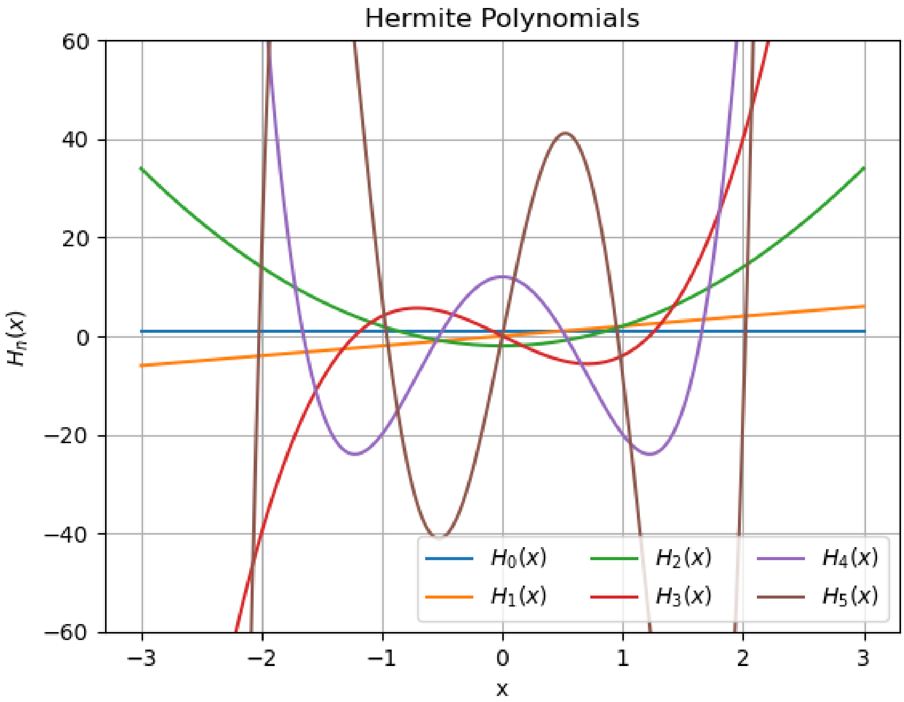
- plotting generated data

In [7]:
```
plt.plot(X,H0,label='$H_0(x)$')
plt.plot(X,H1,label='$H_1(x)$')
plt.plot(X,H2,label='$H_2(x)$')
plt.plot(X,H3,label='$H_3(x)$')
plt.plot(X,H4,label='$H_4(x)$')
plt.plot(X,H5,label='$H_5(x)$')

plt.title("Hermite Polynomials")
plt.xlabel("x")
plt.ylabel("$H_n(x)$")
plt.grid()
plt.legend(loc='lower right',ncol = 3)

ax = plt.gca()
ax.set_ylim(-60,60)

plt.show()
```


Hermite Polynomials
60
40
20
0
−20
−40
−60
$H_n(x)$
−3
−2
−1
0
1
2
3
x
$H_0(x)$
$H_1(x)$
$H_2(x)$
$H_3(x)$
$H_4(x)$
$H_5(x)$