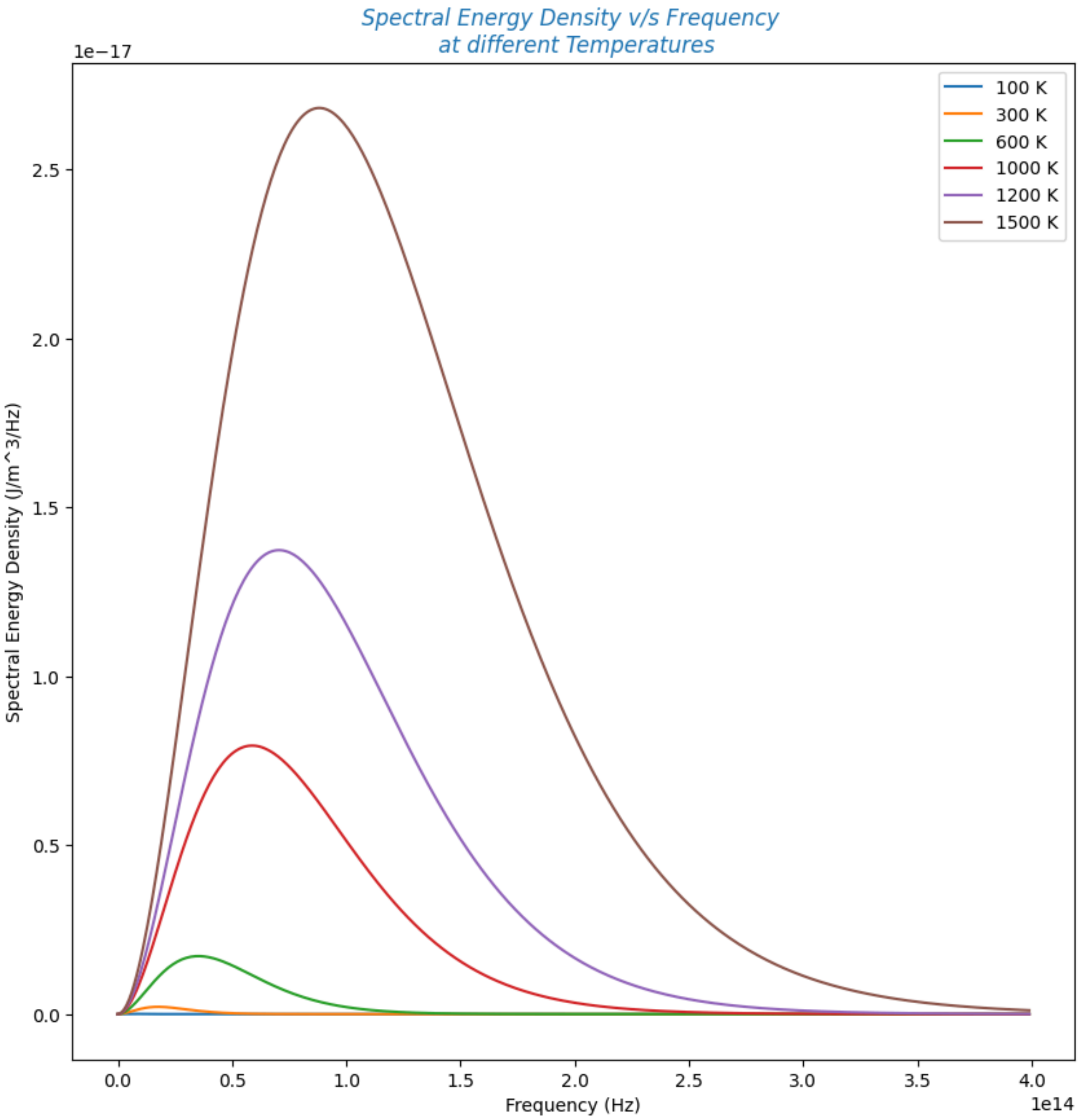
In [18]:

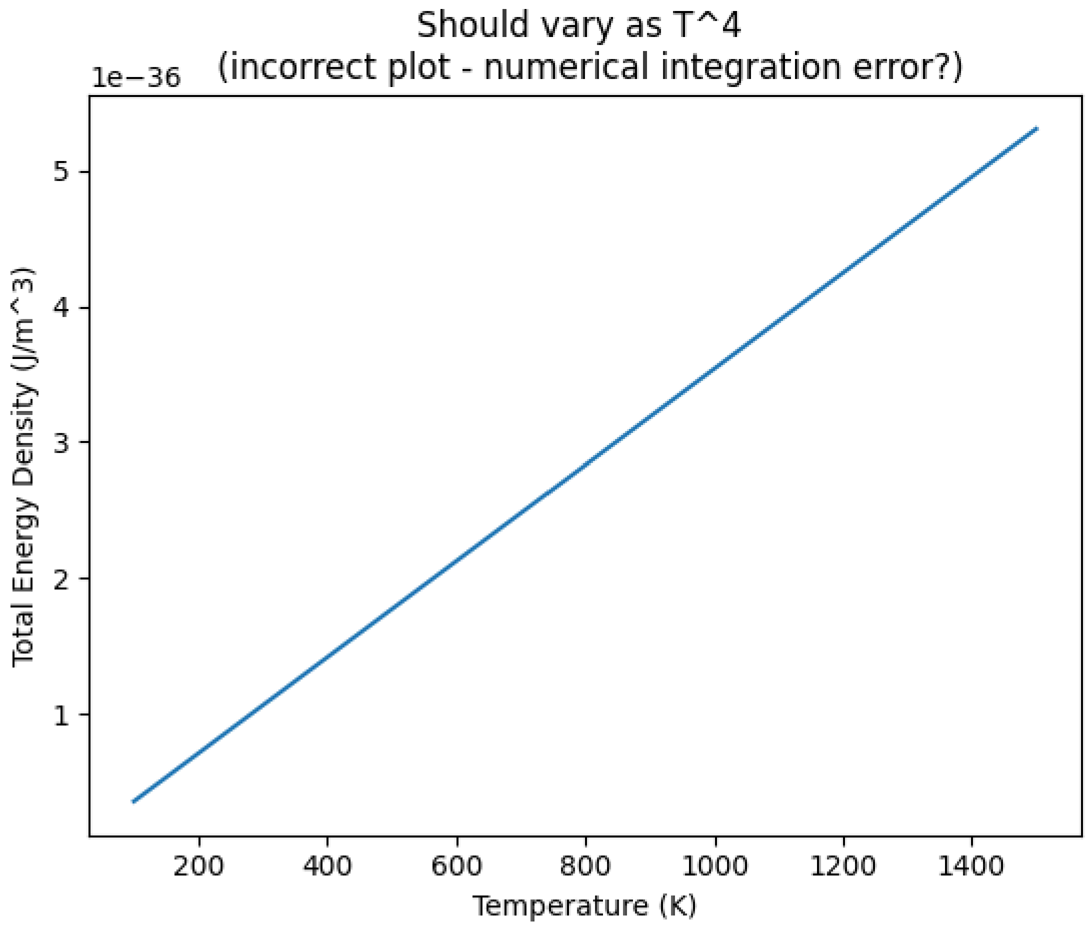
```python
from scipy.constants import pi, c, h, k
from math import e
import matplotlib.pyplot as plt
import numpy as np

#defining the prefactor of spectral energy density in SI units; Joules per metre

rho_0 = (8*pi*h)/(c**3) #value = 6.180644726185984e-58

#defining the constant in the exponent (h/k) in SI units: Kelvin per Hertz

d = h/k #value = 4.799243073366221e-11

#defining the spectral energy density function

def rho(T,v):
    return ((rho_0)*(v**3))*(1/((e**((d*v)/T))-1))
```

In [19]:

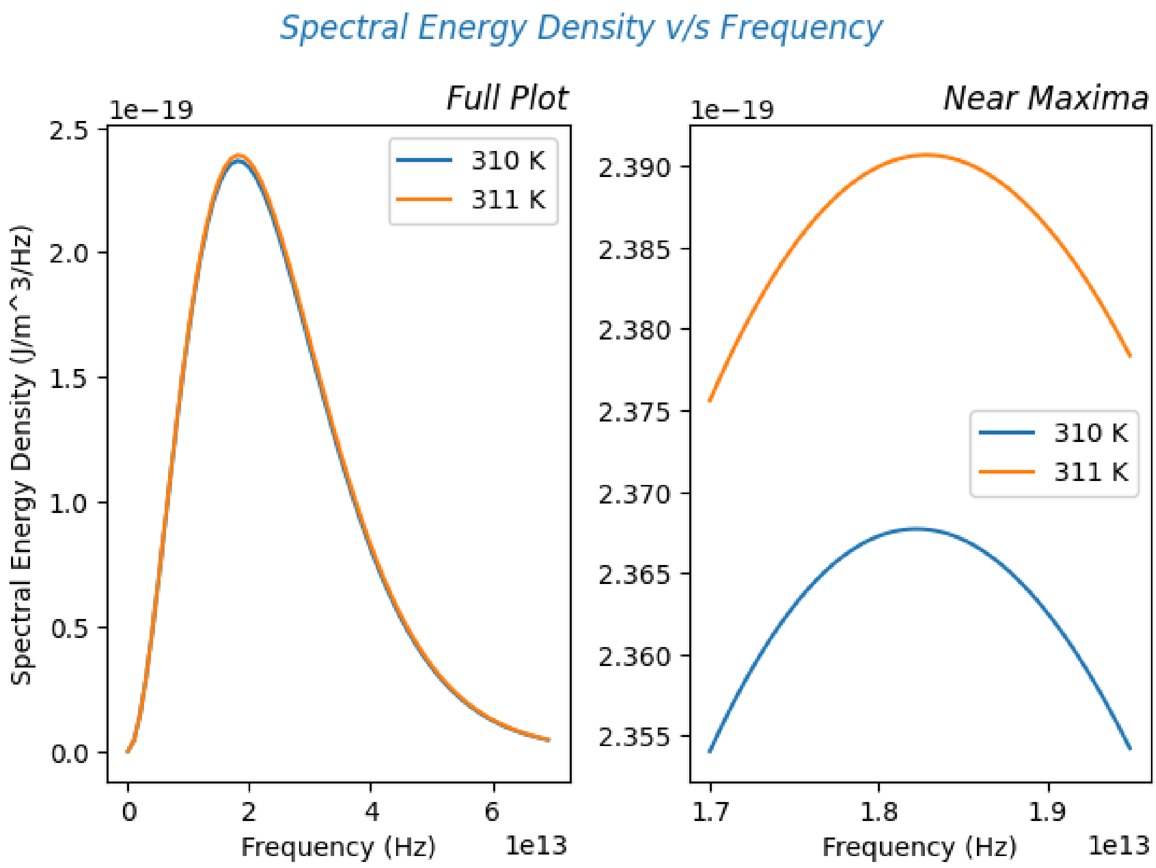
```python
X = np.arange(1,4e14,0.01e14) #frequency points
T1 = 100; T2 = 300; T3 = 600; T4 = 1000; T5 = 1200; T6 = 1500 #according to assi
T = [T1,T2,T3,T4,T5,T6] #storing the temp. values in a list

#plotting the spectral energy density at different temperatures

F = [] #will store the energy density values for each temp. (6 arrays)

for i in range(len(T)):
    f = rho(T[i],X)
    F.append(f.copy())

plt.figure(figsize=(10,10))

for i in range(len(T)): #adding each curve to the plot (6 in total)
    plt.plot(X,F[i])


plt.xlabel("Frequency (Hz)")
plt.ylabel("Spectral Energy Density (J/m^3/Hz)")
plt.title(label="Spectral Energy Density v/s Frequency \n at different Temperatu
plt.legend(["100 K","300 K","600 K","1000 K","1200 K","1500 K"],loc="upper right

plt.show()
```



In [20]:
```
from scipy.integrate import quad #simple numerical integration

Rho = [] #storing the density 'functions' in a list for 6 temperatures (as per a

for i in range(len(T)):
    def Rho_T(v):
        return rho(T[i],v)
    Rho.append(Rho_T)

Tot_Dens = []

for i in range(len(T)):
    I, err = quad(Rho[i],0.01,np.inf)
    Tot_Dens.append(I)

plt.plot(T,Tot_Dens)
plt.xlabel("Temperature (K)")
plt.ylabel("Total Energy Density (J/m^3)")
plt.title("Should vary as T^4 \n (incorrect plot - numerical integration error?)
plt.show() #it gives linear behaviour which is wrong, we know it should vary as
```



In [21]:
```
Tn = 310; Tf = 311 #according to Q5 (n-normal, f-fever)

Xa = np.arange(0.1e12,0.7e14,0.01e14)
Yna = rho(Tn,Xa); Yfa = rho(Tf,Xa)

plt.subplot(1,2,1)
plt.plot(Xa,Yna)
plt.plot(Xa,Yfa)
plt.xlabel("Frequency (Hz)")
plt.ylabel("Spectral Energy Density (J/m^3/Hz)")
plt.legend(["310 K","311 K"])
plt.title(label="Full Plot",loc="right",fontstyle="italic")


Xb = np.arange(1.7e13,1.95e13,0.01e12)
Ynb = rho(Tn,Xb); Yfb = rho(Tf,Xb)

plt.subplot(1,2,2)
plt.plot(Xb,Ynb)
plt.plot(Xb,Yfb)
plt.xlabel("Frequency (Hz)")
plt.legend(["310 K","311 K"])
plt.title(label="Near Maxima",loc="right",fontstyle="italic")

plt.suptitle(t="Spectral Energy Density v/s Frequency",fontstyle="italic",color=
plt.tight_layout()

plt.show()
```



Spectral Energy Density v/s Frequency
Full Plot
Near Maxima
1e−19
Spectral Energy Density (J/m^3/Hz)
310 K
311 K
Frequency (Hz)
1e13
0.0
0.5
1.0
1.5
2.0
2.5
0
2
4
6
1e−19
2.355
2.360
2.365
2.370
2.375
2.380
2.385
2.390
1.7
1.8
1.9
310 K
311 K
Frequency (Hz)
1e13