In [1]:
```
# Declaring constants
interest_rate = 7.1
investment_duration = 25
```

In [2]:
```
# Investing 1.5L at the start of each year before 5th April close
corpus_all_at_start = 0
corpus_all_at_start_by_year = [0]
for i in range(investment_duration):
    interest_i = 0
    investment = 150000
    corpus_all_at_start = corpus_all_at_start + investment
    for j in range(12):
        interest_i = interest_i + corpus_all_at_start*((interest_rate/100)/12)
    corpus_all_at_start = corpus_all_at_start + interest_i
    corpus_all_at_start_by_year.append(corpus_all_at_start)
```

In [3]:
```
# Investing 1.5L at the end of the each year on or after 6th March
corpus_all_at_end = 0
corpus_all_at_end_by_year = [0]
for i in range(investment_duration):
    interest_i = 0
    for j in range(12):
        interest_i = interest_i + corpus_all_at_end*((interest_rate/100)/12)
    corpus_all_at_end = corpus_all_at_end + interest_i
    investment = 150000
    corpus_all_at_end = corpus_all_at_end + investment
    corpus_all_at_end_by_year.append(corpus_all_at_end)
```

In [4]:
```
# Investing 12500 each month before 5th close
corpus_sip_at_start = 0
corpus_sip_at_start_by_year = [0]
investment_sip = 12500
for i in range(investment_duration):
    interest_i = 0
    for j in range(12):
        corpus_sip_at_start = corpus_sip_at_start + investment_sip
        interest_i = interest_i + corpus_sip_at_start*((interest_rate/100)/12)
    corpus_sip_at_start = corpus_sip_at_start + interest_i
    corpus_sip_at_start_by_year.append(corpus_sip_at_start)
```

In [5]:
```
# Investing 12500 each month on or after 6th
corpus_sip_at_end = 0
corpus_sip_at_end_by_year = [0]
investment_sip = 12500
for i in range(investment_duration):
    interest_i = 0
    for j in range(12):
        interest_i = interest_i + corpus_sip_at_end*((interest_rate/100)/12)
        corpus_sip_at_end = corpus_sip_at_end + investment_sip
    corpus_sip_at_end = corpus_sip_at_end + interest_i
    corpus_sip_at_end_by_year.append(corpus_sip_at_end)
```

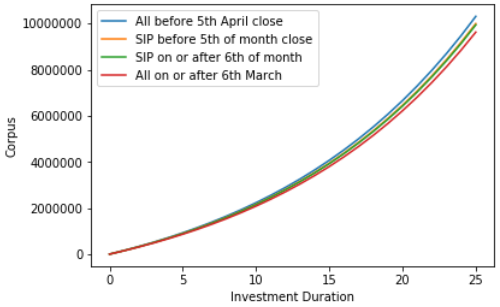
In [6]:
```
# Graphing the corpus over the years
%matplotlib inline
import matplotlib.pyplot as plt

investment_duration_by_year = []
for i in range(0,investment_duration+1):
    investment_duration_by_year.append(i)

plt.plot(investment_duration_by_year, corpus_all_at_start_by_year, label="All before 5th April close")
plt.plot(investment_duration_by_year, corpus_sip_at_start_by_year, label="SIP before 5th of month close")
plt.plot(investment_duration_by_year, corpus_sip_at_end_by_year, label="SIP on or after 6th of month")
plt.plot(investment_duration_by_year, corpus_all_at_end_by_year, label="All on or after 6th March")
plt.legend()

ax = plt.gca()
ax.set_xlabel('Investment Duration')
ax.set_ylabel('Corpus')
ax.ticklabel_format(useOffset=False, style='plain')

plt.show()
```



In [7]:
```
# Comparing values for different scenarios
from tabulate import tabulate
table_rows = []
for i in range(1,investment_duration+1):
    table_rows.append([i, round(corpus_all_at_start_by_year[i], 2), round(corpus_sip_at_start_by_year[i],2), round(corpus_sip_at_end_by_year[i], 2), round(corpus_all_at_end_by_year[i], 2)])

print(tabulate(table_rows, headers=["Year", "All before 5th April close", "SIP before 5th close", "SIP on or after 6th", "All on or after 6th March"], disable_numparse=True, tablefmt="grid", stralign="center"))
```

Year	All before 5th April close	SIP before 5th close	SIP on or after 6th	All on or after 6th March
1	160650.0	155768.75	154881.25	150000.0
2	332706.15	322597.08	320759.07	310650.0
3	516978.29	501270.22	498414.21	482706.15
4	714333.75	692629.16	688682.87	666978.29
5	925701.44	897574.58	892460.61	864333.75
6	1152076.24	1117071.13	1110706.56	1075701.44
7	1394523.66	1352151.93	1344447.97	1302076.24
8	1654184.84	1603923.46	1594785.03	1544523.66
9	1932281.96	1873570.78	1862896.02	1804184.84
10	2230123.98	2162363.05	2150042.88	2082281.96
11	2549112.78	2471659.58	2457577.18	2380123.98
12	2890749.79	2802916.16	2786946.41	2699112.78
13	3256643.02	3157691.96	3139700.85	3040749.79
14	3648514.68	3537656.84	3517500.87	3406643.02
15	4068209.22	3944599.22	3922124.68	3798514.68
16	4517702.07	4380434.52	4355476.78	4218209.22
17	4999108.92	4847214.12	4819596.88	4667702.07
18	5514695.66	5347135.07	5316669.51	5149108.92
19	6066889.05	5882550.41	5849034.29	5664695.66
20	6658288.17	6455980.24	6419196.98	6216889.05
21	7291676.63	7070123.59	7029841.21	6808288.17
22	7970035.67	7727871.11	7683841.19	7441676.63
23	8696558.2	8432318.71	8384275.16	8120035.67
24	9474663.84	9186782.09	9134439.95	8846558.2
25	10308014.97	9994812.37	9937866.44	9624663.84

In [ ]: