Activity Guide # 1

CODE:

```python
#Item no.1: Bubble Sort in Ascending Order
data1 = [23, 89, 7, 56, 44]
print("1.")
print("data 1 before Bubble Sort(Ascending Order):", data1)
n = len(data1)
for i in range(n):
        for j in range(0, n - i - 1):
            if data1[j] > data1[j + 1]:
                data1[j], data1[j + 1] = data1[j + 1], data1[j]
print("Bubble Sort(Ascending Order):", data1)
print()

#Item no.2: Insertion Sort in Ascending Order
data2 = [12, 78, 91, 34, 62]
print("2.")
print("data2 before Insertion Sort(Ascending Order):", data2)
for i in range(1, len(data2)):
    key = data2[i]
    j = i - 1
    while j >= 0 and key < data2[j]:
        data2[j + 1] = data2[j]
        j -= 1
        data2[j + 1] = key
print("Insertion Sort(Ascending Order):", data2)
print()

#Item no.3: Selection Sort in Descending Order
data3 = [5, 99, 48, 15, 67]
print("3.")
print("data3 before Selection Sort(Descending Order):", data3)
for i in range(len(data3)):
    max_idx = i
    for j in range(i + 1, len(data3)):
        if data3[j] > data3[max_idx]:
            max_idx = j
    data3[i], data3[max_idx] = data3[max_idx], data3[i]
print("Selection Sort(Descending Order):", data3)
print()
```

```python
#Item no.4: Insertion Sort in Descending Order
data4 = [38, 82, 25, 74, 13]
print("4.")
print("data4 before Insertion Sort(Descending Order):", data4)
for i in range(1, len(data4)):
    key = data4[i]
    j = i - 1
    while j >= 0 and key > data4[j]:
        data4[j + 1] = data4[j]
        j -= 1
        data4[j + 1] = key
print("Insertion Sort(Descending Order):", data4)
print()

#Item no.5: Copy the values from the second and third index of the previous datasets into a new dataset
print("5.")
#Function to perform Selection Sort in Ascending Order
1 usage  new *
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
    return arr
new_data = [89, 7, 78, 91, 99, 48, 82, 25]
ascending_order = bubble_sort(new_data.copy())
print("Merged Dataset(Ascending Order):", ascending_order)

#Function to perform Selection Sort in Descending Order
1 usage  new *
def selection_sort_descending(arr):
    n = len(arr)
    for i in range(n):
        max_idx = i
        for j in range(i + 1, n):
            if arr[j] > arr[max_idx]:
                max_idx = j
        arr[i], arr[max_idx] = arr[max_idx], arr[i]
    return arr
```

```python
descending_order = selection_sort_descending(new_data.copy())
print("Merged Dataset(Descending Order):", descending_order)
print()

#Item no.6: Create a list with values from item 1 to 4 and implementing Selection Sort(Ascending Order)
print("6.")
data_combined = [23, 89, 7, 56, 44, 12, 78, 91, 34, 62, 5, 99, 48, 15, 67, 38, 82, 25, 74, 13]
for i in range(len(data_combined)):
    min_idx = i
    for j in range(i + 1, len(data_combined)):
        if data_combined[j] < data_combined[min_idx]:
            min_idx = j
    data_combined[i], data_combined[min_idx] = data_combined[min_idx], data_combined[i]
print("Selection Sort(Ascending Order):", data_combined)
print()

#Item no.7: Printing even and odd values of the list from item number 6
print("7.")
1 usage  new *
def seperate_even_odd(arr):
    evens = []
    odds = []
    for num in arr:
        if num % 2 == 0:
            evens.append(num)
        else:
            odds.append(num)
    return evens, odds

#Example Dataset:
data_combined = [23, 89, 7, 56, 44, 12, 78, 91, 34, 62, 5, 99, 48, 15, 67, 38, 82, 25, 74, 13]
#Call the functions to seperate even and odd numbers
evens, odds = seperate_even_odd(data_combined)
print("Even Numbers:", evens)
print("Odd Numbers:", odds)
```

OUTPUT:

```
1.
data 1 before Bubble Sort(Ascending Order): [23, 89, 7, 56, 44]
Bubble Sort(Ascending Order): [7, 23, 44, 56, 89]

2.
data2 before Insertion Sort(Ascending Order): [12, 78, 91, 34, 62]
Insertion Sort(Ascending Order): [12, 34, 62, 78, 91]

3.
data3 before Selection Sort(Descending Order): [5, 99, 48, 15, 67]
Selection Sort(Descending Order): [99, 67, 48, 15, 5]

4.
data4 before Insertion Sort(Descending Order): [38, 82, 25, 74, 13]
Insertion Sort(Descending Order): [82, 74, 38, 25, 13]

5.
Merged Dataset(Ascending Order): [7, 25, 48, 78, 82, 89, 91, 99]
Merged Dataset(Descending Order): [99, 91, 89, 82, 78, 48, 25, 7]

6.
Selection Sort(Ascending Order): [5, 7, 12, 13, 15, 23, 25, 34, 38, 44, 48, 56, 62, 67, 74, 78, 82, 89, 91, 99]

7.
Even Numbers: [56, 44, 12, 78, 34, 62, 48, 38, 82, 74]
Odd Numbers: [23, 89, 7, 91, 5, 99, 15, 67, 25, 13]

Process finished with exit code 0
```