

First Name: _____

Last Name: _____

C++ Lab 3 – Passing Parameters

Considering the next given statements

```
int sum(int x, int y)
{
    x = x + y;
    printf("\n Inside function -> x=%d", x);
    printf("\n Inside function -> y=%d", y);
    return x;
}
```

analyze the next lines

	Expected value/error	Actual result/error	Reason for heaving an error
printf("\n Sum = %d", sum(5, 3));			
int x = 5; int y = 3; printf("\n Sum = %d", sum(x, y)); printf("\nOutside function -> x=%d", x); printf("\nOutside function -> y=%d", y);			

Modify the sum function like this

```
int sum(int& x, int y)
{
    x = x + y;
    printf("\nInside function -> x=%d", x);
    printf("\nInside function -> y=%d", y);
    return x;
}
```

	Expected value/error	Actual result/error	Reason for heaving an error
printf("\n Sum = %d", sum(5, 3));			
int x = 5; int y = 3; printf("\n Sum = %d", sum(x, y)); printf("\nOutside function -> x=%d", x); printf("\nOutside function -> y=%d", y);			
printf("\n Sum = %d", sum(&x, y));			

Modify the sum function again

```
int sum(int* x, int y)
{
    (*x) = (*x) + y;
    printf("\nInside function -> x=%d", *x);
    printf("\nInside function -> y=%d", y);
    return *x;
}
```

	Expected value/error	Actual result/error	Reason for heaving an error
printf("\n Sum = %d", sum(5, 3));			
int x = 5; int y = 3; printf("\n Sum = %d", sum(x, y)); printf("\nOutside function -> x=%d", x); printf("\nOutside function -> y=%d", y);			
printf("\n Sum = %d", sum(&x, y));			

Consider the following functions

```
void sort(int* v, int n)
{
    for (int i = 1; i<n; ++i)
    {
        for (int j = 0; j<(n - i); ++j)
            if (v[j]>v[j + 1])
            {
                int temp = v[j];
                v[j] = v[j + 1];
                v[j + 1] = temp;
            }
    }
}
```

```
void print_vector(int* vector, int n) {
    printf("\n vector = ");
    for (int i = 0; i < n; i++)
        printf("%d ", vector[i]);
}
```

```
int* elemente = new int[5] { 3, 1, 4, 5, 2 };
```

	Expected value/error	Actual result/error	Reason for heaving an error

POO Handout – Lab 3

<pre>sort(elemente, 5); print_vector(elemente, 5);</pre>			
<pre>sort(&elemente, 5);</pre>			
<pre>sort(*elemente, 5);</pre>			
<pre>sort(elemente[2], 5);</pre>			
<pre>int a = 2; int* p = &a; sort(p, 1);</pre>			

Add the following line as the first one in the sort function

```
v = new int[n];
```

	Expected value/error	Actual result/error	Reason for heaving an error
<pre>sort(elemente, 5); print_vector(elemente, 5);</pre>			

Now change the function's signature

```
void sort(int*& v, int n)
{
    ...
}
```

	Expected value/error	Actual result/error	Reason for heaving an error
<pre>sort(elemente, 5); print_vector(elemente, 5);</pre>			

What about this?

```
void sort(int** v, int n)
{
    (*v) = new int[n];
    for (int i = 1; i<n; ++i)
    {
        for (int j = 0; j<(n - i); ++j)
            if ((*v)[j]>(*v)[j + 1])
            {
                int temp = (*v)[j];
                (*v)[j] = (*v)[j + 1];
                (*v)[j + 1] = temp;
            }
    }
}
```

	Expected value/error	Actual result/error	Reason for heaving an error
sort(elemente, 5); print_vector(elemente, 5);			
sort(&elemente, 5); print_vector(elemente, 5);			