# CSI 250 Programming II - Exam I

## Exercise #1:

Write C++ statements that perform the specified task.
Given: Floating point numbers are stored in 8 bytes.
The starting address of the array is 1048500 in RAM.

**a.** Declare an array of type float called “values” with 10 elements:
0.0, 1.1, 2.2, 3.3, ... 9.9
Given: const int SIZE = 10;

```
float values [ SIZE ] = {0.0, 1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7,
8.8, 9.9};
```

**b.** Declare a pointer vPtr that points to a variable of type float:

```
float *vPtr;
```

**c.** Use a for statement to print the elements of the array “values” using array subscript notation:

```
for (int i = 0; i < SIZE; i++)
     cout << values[ i ] << endl;
```

**d.** Assign the starting address of array “values” to vPtr in 2 ways:

```
1- vPtr = values;
2- vPtr = &values[0];
```

**e.** Use a for statement to print elements of array “values” using:

1- Pointer/Offset notation with the pointer:

```
for (int i = 0; i << SIZE; i++)
     cout << *(vPtr + i) << endl;
```

2- Pointer/Offset notation with the array name:

```
for (int i = 0; i << SIZE; i++)
     cout << *(values + i) << endl;
```

3 - Pointer/Subscript notation with the pointer:

```
for (int i = 0; i << SIZE; i++)
     cout << vPtr [ i ] << endl;
```

4- Array/Subscript notation with the array:

```
for (int i = 0; i << SIZE; i++)
     cout << values [ i ] << endl;
```

**f.** Refer to the 4th element of “values” using all the notations mentioned in part “**e**”

1- `values [ 3 ]`
2- `*(values + 3)`
3- `vPtr [ 3 ]`
4- `*(vPtr + 3)`

**g.** vPtr points to the beginning of “values”. What address is referenced by (vPtr + 8)? What values is stored in that location?

Address: `1048564`
Value Stored: `8.8`

**Exercise #2:**

Create a class “Rectangle” with attributes “length” and “width” which default to “1”. Provide member functions that calculate the rectangle’s “Perimeter” and “Area”.
Also, provide access methods (set and get functions) for the attributes.
The set functions should verify “length” and “width” are floating point numbers greater than 0.0 and less than 20.0.

Provide a driver in which you:

- Create automatic object “r1” with initial values of width = 15, length = 17 then print the Perimeter and Area.
- Create dynamic object “r2” with initial values of width = 10, length = 14 then print the Perimeter and Area.
- At the destruction of each object, display the message “Time to leave”.
- Make sure that nothing remains unnecessarily allocated in memory when the program terminates.