COMP710: Studio Session 02 – Exercise:

EXERCISE NAME: *C++ – Using Pointers and the Free Store*

Add a new C++ Project named "Using Pointers and the Free Store" to your "SS02" Visual Studio Solution for this exercise.

Declare a structure which represents an NPC (Non-Player Character) in a 2D game. Add fields to this structure to represent the follow details: strength, health, tiredness, position and whether the NPC is alive or not.

Next declare a function called **`CreateNPC()`** which takes no parameters and returns a pointer to an **`NPC`** structure.

In the **`CreateNPC`** function, allocate an **`NPC`** structure on the Free Store. Then assign the fields of the NPC random values based upon the following criteria:

- 3 < Strength < 20
- 50 < Health < 200
- 1 < Tiredness < 15
- 0 < Position X < 100
- 0 < Position Y < 100
- Alive is true

Then return this allocation to the calling function.

Write another function called **`PrintNPCDetails()`** which takes in a **`NPC`** pointer parameter, and prints the details of the **`NPC`** in the following format:

```
NPC's Current State:
 - Position: (0, 0)
 - Health: 112
 - Strength: 15
 - Tiredness: 10
 - Alive: Yes
```

In the **`main`** function, create an array of 10 **`NPC`** pointers.

Set each pointer in the array to be null using a loop.

Next, call **`CreateNPC`** ten times and store the resulting allocations in the **`NPC`** pointer array.

Finally, call **`PrintNPCDetails`** ten times, each time with address of an **`NPC`** stored on the Free Store.

The following is an example of this:

```
/* #include required headers here */

struct NPC
{
    /* Insert your code here */
};

/* Insert Prototypes here */

int main()
{
    // Declare NPC pointer array.

    // Call CreateNPC ten times, store the results.

    // Call PrintNPCDetails for each NPC in the array.

    return 0;
}

/* Insert function definitions here */
```

Before the program finishes, deallocate the Free Store allocations, and hence avoid a memory leak occurring!

Once complete, commit your program's source code to your individual SVN folder – include the **`.sln`**, **`.vcxproj`** and **`.cpp`** files, and ensure you do not commit any build output files.