COMP710: Studio Session 03 – Exercise:

EXERCISE NAME: *C++ – Using an Object Factory*

Add a new C++ Project named "Using an Object Factory" to your "SS03" Visual Studio Solution for this exercise.

Create four new classes in this project:

- **`Entity`**
- **`Zombie`**
- **`HealthPack`**
- **`AmmoPack`**

Ensure each of the four classes has their own header file (**`.h`**) and a source file (**`.cpp`**).

1) Ensure **`Zombie`**, **`HealthPack`** and **`AmmoPack`** are all subclasses of **`Entity`**. The **`Entity`** class must be abstract.

2) Give each class an appropriate constructor and destructor.

3) Add a Boolean member of type **`bool`** to the **`Entity`** class named **`m_bVisible`**. Declare a member method **`IsVisible`** in the **`Entity`** class, which returns a **`bool`** based upon the member data named **`m_bVisible`**.

4) Add a file named **`main.cpp`**. In this file declare and define a **`main`** function. In the **`main.cpp`** file, declare an enumeration called **`EntityType`** with three elements, **`ZOMBIE`**, **`HEALTHPACK`**, and **`AMMOPACK`**.

5) Add a simple object factory function with the signature:

   ```
   Entity* CreateEntity(EntityType entityType);
   ```

   … which takes in an enumerated value parameter representing the type of **`Entity`** to create. This function must allocate the appropriate **`Entity`** on the Free Store using **`new`**, and then return to the caller the pointer to the newly created **`Entity`** instance.

6) In the **`main`** function, call **`CreateEntity`** with different values of **`EntityType`**. Store the pointer returned in the **`main`** function, and then query each entity instance by calling the **`IsVisible`** method that belong to the **`Entity`** via the pointer.

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