Homework 4

Question 1 — Object Pooling

Why is Object Pooling commonly used in Unity3D?

A) To reduce the need for prefabs in a project
B) To improve performance by reusing inactive GameObjects instead of destroying and instantiating them repeatedly
C) To automatically optimize GPU rendering
D) To increase garbage collection frequency

Answer:________

Question 2 — Using a Factory to Create Objects

What is the main benefit of using a Factory Pattern to create objects (e.g., from prefabs) in Unity?

A) It stores all object instances in a single static variable
B) It hides object creation details, allowing flexible and consistent initialization of prefabs
C) It automatically adds colliders to every object created
D) It makes all objects persist across scenes by default

Answer:________

Question 3 — Interfaces & Generics in C# / Unity

Why might a Unity developer use multiple interfaces and generics in C#?

A) To inherit multiple base classes simultaneously
B) To make code more rigid and type-specific
C) To promote code reusability, flexibility, and type safety while following clean architecture principles
D) To increase compile times for stronger performance

Answer:________

4. Coding Question (Bonus)

Unity3D Coding Challenge: Implement Your Own Object Pool

Objective:
Create a simple **object pooling system** in Unity to efficiently manage cube objects that spawn, fall onto a plane, and get recycled when they leave the scene.

🧱 **Requirements:**

1. **Create a Plane:**
   - Acts as the ground (use Unity's built-in Plane).
   - Tag or name it `"Ground"` for easy reference.
2. **Create a Cube Prefab:**
   - A standard Unity Cube with a Rigidbody component for gravity.
   - Save it as a prefab in your Assets folder.
3. **Implement an `ObjectPool<T>` class:**
   - Manages a list or queue of inactive objects.
   - Provides methods:
   - `public T GetFromPool();`
   - `public void ReturnToPool(T obj);`
   - When the pool is empty, instantiate a new cube.
4. **Spawner Script:**
   - Periodically spawns cubes above the plane using your pool:
   - `InvokeRepeating("SpawnCube", 1f, 1f);`
   - Use random X/Z positions within a defined range.
5. **Cube Behavior Script:**
   - Detect when a cube falls below the plane (e.g., `y < -5f`).
   - When that happens, deactivate the cube and return it to the pool.
6. **Test Scene Setup:**
   - Create an empty GameObject named `"GameManager"`.
   - Attach your `Spawner` script and assign references (plane, cube prefab).
   - Press **Play** — cubes should spawn, fall, disappear off-screen, and be reused.