Logical Operators

- Iteration statements are critical to a functioning loop. Iteration statements result in the statements within the iteration are executed repeatedly. The amount of times the code is looped depends on the iteration statement parameters.
- For loops are one version of iteration statements. These loops contain particular clauses that change the initialization and iteration of a looped code block. Other than that, they are the same as while loops. The clauses include the init, condition, and iteration clauses.
- public List<GameObject> wrongAssignment;

  void OnTriggerEnter(Collider _assignmenet)

  {

  wrongAssignment.Add(_assignment.gameObject);

  }
- Another type of loop is the “foreach” loop. This iteration loops through each element in an enumerable object. This is useful, as almost every type in C# that are used to represent a set or list of elements are enumerable, like strings, arrays, and lists.
- public char[] hello = {‘h’, ‘e’, ‘y’, ‘d’, ‘u’, ‘d’, ‘e’};

  public void HeyDude (char c) {

  foreach (char c in hello) {

  print(c); //heydude

  }

  }
- public int o;

  string[] numbers = {“one”, “two”, “three”};

```
void goCounting (char c) {
        for (int o, o<3, ++o) {
                print(numbers[c])
                print(o);
        }
```

- public List<int> countYes;

```
int i = 8;
void OnTriggerEnter(Collider _integer) {
        while (i<12){
                countYes.Add(_integer.int);
        }
}
```

- Lists are very similar to arrays. They retain multiple variables in specific orders. The main difference between lists and arrays is the dynamic, fluid nature of lists, as opposed to the rigidness of arrays. Arrays have a fixed number of elements, but lists do not.
- public int[] collegeJoke = new int[3];

```
public int i;
public void Start () {
        collegeJoke[0] = 000;
        collegeJoke[1] = 020;
        collegeJoke[2] = 400;
}
public void collegeJoke () {
```

```
        int i == (collegeJoke[0] + collegeJoke[1] + collegeJoke[2])
        print(i + “blaze”);
    }
```

- ```
  public bool canDrink = false;
  int age = 23;

  void Start () {
      if(age > 21) {
          canDrink = true;
          StartCoroutine(GoDrink());
      } else {
          print(“you aren’t 21 yet silly”);
      }
  }

  IEnumerator GoDrink () {

      while (canDrink) {
          print(“Drinking”);
          yield return new WaitForSeconds(0.1f);
      }
      print(“staying legal”);
  ```
  ```

```
}
```