# Lab 6 Handling Multiple Objects with a Collection

## Learning Objectives

* Creating instance objects and storing in an ArrayList (Collection)
* Iterating through a collection of objects

Ex1 Using an ArrayList of Objects

In a new sketch add the code below

ArrayList<IntObject> intArrayList; //declare an ArrayList of class IntObject

void setup()

{

intArrayList; = new ArrayList<IntObject>(); //create an empty list

//add numbers 1 to 10

IntObject numberObject;

for (int number=1; number<=10; number++)

{

numberObject = new IntObject(number);

intArrayList.add(numberObject);

}

}

And add the class below to a new tab

class IntObject

{

int value;

IntObject(int number)

{

value = number;

}

}

What does it do?

Comment your code appropriately.

Ex 1.A. How would we print to console all the numbers in the ArrayList? Add a suitable **procedure** and test it.

Alter the code so that the arrayList contains the numbers 16,14,12,10,8 in that order.

Ex 1.B How would we sum all the numbers in the ArrayList and print to console? Add a suitable **function** that will **return** the total and test it. We covered functions in Week 2 on moodle.

Ex 1.C Write a **Boolean function** that will search for a specified value in the ArrayList and return true if it’s there, false otherwise. Test it.

Ex 2. **Simple collecting game**

We’ll write a simple game where we can move a green circle around the screen (use the mouse) to collect (remove) multiple red squares.

Top down design what do we need? What Classes? What Events?

Ex 3. Extension exercise : try to incorporate these ideas in to your Defenderz code from week 5.

**Instructions**: We are going to add an **ArrayList collection** to handle multiple Alien objects, in the Defenderz program you were working on previously (part 2 this week).  First ensure those exercises are completed up to and including :

Complete the 1st prototype of the game

·         Add 3 or more aliens (2 more), see lecture powerpoint for use of a collection.

·         Use a **gameMode** variable to stop the game if a crash occurs (we covered this idea with the racing Motorbikes lab exercise – see “Objects, Using a Class” (on moodle).

* **alter your crash function** method to use a distance measure (between the two objects) rather than being colour based. This is so we know which alien was hit, the alien should be passed as a parameter.

·         Allow the alien to move up or down in some sequence (as well as left). For example a random step up or down (add a random number between -5 and +5 to y).

·         If the alien goes off the screen a **new** alien should appear on the right

Implement an **arrayList** of your Alien class and add 5 or more instances to the arrayList, each with a random  value ( <https://processing.org/reference/random_.html>) for y and possibly x (somewhere near the right edge of the screen or just off it).

You should remove an Alien object from the arrayList when it goes off the left of the screen

Add a new alien to the arrayList

 The powerpoint slides and lecture recording will help you manage this task.