# Web 5\_B. Developing a Class (animated objects)

## Learning Objectives

* Using multiple instances of **PImage**
* Designing and implementing a Class
* Writing methods
* Writing a constructor

In your Programming directory create a directory and save your code after each exercise.

Ex1. Save the code below as **walkers.pde**. On moodle you will find a zip file of **gif images** for a walking person sequence. Download these and place the unzipped images in your **walkers** directory alongside the code file.

We will create a suitable class to allow multiple people to walk across the screen at differing speeds. The image sequence should be used as movement takes place. When a person reaches the right edge of the screen they should restart at the left edge (wrap around).

PImage img1;

int x=50,y=50;

size(400,400);

img1 = loadImage("walk1.gif"); //loads from .pde source code directory

image(img1,x,y);

**Design**

* What **members** will our Walker class require?
* Constructor?
* Methods?

**Ex1.B –** On moodle (walkerBug) is a bug-ridden solution to problem above – get it working

**Exercise 2. Directional Objects**. Using a drawing package create 4 simple images for an animated object. The easiest technique, is to draw the first, save it, then modify it and save repeatedly. Most drawing packages will allow you to mirror an image file (Paint 3D is in installed in the labs, <https://www.sumopaint.com/home/#app>, is an online package that works well with **google chrome**).

One image for when the Object is heading each of:

* up and right
* up and left
* down and right
* down and left

1. Create a class that allows multiple objects to move in **all 4** directions around the screen whilst using your 4 images. The class should contain a **constructor**, a **render** and a **move** method (perhaps more).
2. Create a program that allows 3 instance objects of your class to move around the screen, bouncing off each other (e.g. see lecture notes – ***crash function*** method in Bird class) and off the edges.

When detecting collisions between objects, it’s easier if the **x,y** location of each object is in the **centre** rather than the top left corner. The command “ **imageMode(CENTER);**” can be added to **setup()** and will cause every PImage drawn to the screen to be drawn with x,y at the centre, rather than top left.

[note : if your objects seem to collide and stick together – it’s likely that you are detecting a collision but when they move apart (next move()) a collision is detected again which repeats – objects keep changing direction. Try reducing the value you are using for the distance]

See marking criteria.

**Extension exercise** : when your objects bounce off each other, attempt to draw a crash sequence (e.g. explosion) which in its simplest terms could just be a growing set of concentric circles in different colours. Hint look back at our use of a **gameMode** variable in the last lab.

Code for Ex1.b