CS1013 Programming Project
Exercise 5

1. Building on the ***Alien***, ***Player*** and ***Bullet*** classes from previous weeks, add a ***Bomb*** class:

- the constructor ***Bomb()*** will create a bomb at a particular x, y position given as arguments.
- the ***move()*** method will move the bomb one pixel down the screen.
- the ***draw()*** method will draw the bomb at its current position
- an **offScreen()** function which will return a boolean value - **true** if the bomb has gone past the bottom of the screen, **false** if it is still onscreen.

Demonstrate your code by creating a single instance of Bomb somewhere at the top of the screen and have it move down the screen. Aliens, player and bullet should behave as before. (30 marks)

2. Add a ***collide()*** method to **Bomb** which will check whether the bomb has collided with the player (it takes an instance of Player as an argument). The method should return a boolean value - **true** if there has been a collision, **false** if there has not been a collision. Alter the main draw() method so that if the bomb has collided with the player, a "game over" message should be displayed (all messages should be drawn to the screen using PFont methods).

Demonstrate your code by having the bomb go off the bottom of the screen without colliding, and also colliding with the player (with "game over" message). (30 marks)

3. Extend your program so that the aliens **that are alive** drop bombs. Each Alien will drop at most one bomb at a time. Once a bomb has gone off the bottom of the screen, the alien can drop another bomb (if it is alive, exploding or dead aliens cannot drop bombs).

One approach is to give each Alien a variable of type ***Bomb***. As part of the move method, the Alien can decide to drop a bomb (ie. create a **new Bomb**), or if it has already got a bomb, it can ***move()*** the bomb. A ***getBomb()*** method returns the bomb if the Alien has one (ie. it is not **null**). This ***getBomb()*** method in Alien is used in the main ***draw()*** method to check whether any of the bombs have collided with the player.

You might have something like the following inside of the loop which moves all the aliens:

```
Bomb aBomb = theAliens[i].getBomb();
if (aBomb!= null) {
   if (aBomb.collide(thePlayer))
         // bad news for player
    }
```

Demonstrate your code showing that only the aliens who are alive drop bombs. As before, bombs that do not hit the player should go off the bottom of the screen without colliding, if they collide with the player there should be a "game over" message. If the player kills all the aliens you should print a “You Win” message to the screen (20 Marks)

4. Add shields that the player can hide behind. Alien bullets can damage the shield, but so can the player bullets. The damage to the shield must be gradual, and must be represented visually. (20 Marks)

Lab Challenge (Don’t worry if you can’t do this) Use the processing sound library or minim library to add sounds, using a combination of recordings, oscillators and effects (you must use all three). You will need headphones, a laptop with integrated speakers, or small speakers to demonstrate.