

## COMP1006/1406 – Summer 2016

Submit a single file called `assignment5.zip` to cuLearn.  
There are 100 marks possible marks. The assignment it out of 100.

### Capture the Flag

This assignment is meant to help you with your project. Before starting this assignment, download and read the Project specification on the course webpage. In this assignment, you will implement several player classes that will help you create the player classes needed for the project. These classes will not be directly used in the project but most of the functionality (behaviour) will be needed in the project.

#### 1: Compiling [0 marks]

Download the [project-files.zip](#) archive from the course webpage. Find and fix the bug so that it compiles and runs.

This is worth zero marks but is essential for your project and to be used to help you with the rest of this assignment. Grading of the assignment and the project will be partly based on watching the game play (or parts of the game play). Make sure your code is visually doing what it is supposed to be doing.

#### 2: Stopping Player [20 marks]

All players need to stay within the field boundaries. This first class you write will lay the foundation for the logic needed in all your players.

Create a class called `Stopping` that extends the `Player` class. A Stopping player will initially move in a random direction and stops moving when it reaches the border of the playing field (without leaving the playing field). The direction that a player moves in is determined by its speed in each direction: `speedX` and `speedY`. You will add this logic to manipulate the speed attributes for stopping in its `play()` method.

Take care to notice that the images of the players have some size to them. Your player should not “visually” leave the field. All parts of the image should be on or inside the field boundaries.

Mark breakdown: 20 marks for correctness

Put your `Stopping.java` files in your `assignment5.zip` file.

### 3: Seeker Player [20 marks]

The next step is to allow your player to move in a specified direction. Create a class called `Seeker` that extends the `Player` class.

Your Seeker will find the flag of the opposing team. The `Field` class has methods called `getFlag1Position()` and `getFlag2Position()` that each returns an array of two integers that represent the x and y coordinates of where the given flag is located on the field, respectively. Your player should stop moving when it calls the `pickUpFlag()` method and that method returns `true`.

We will run your program with different locations for the flags. Do not hardcode your player using the flag locations in the provided `CaptureTheFlag.java` file.

Mark breakdown: 20 marks for correctness

Put your `Seeker.java` file in your `assignment5.zip` file.

### 4: Random Walker Player [20 marks]

Create a class called `RandomWalker` that extends the `Player` class.

Your `RandomWalker` player will continually walk in the playing field. When it reaches a border of the playing field, it should “bounce” off the wall and continue walking and bouncing off the borders.

Mark breakdown: 20 marks for correctness

Put your `RandomWalker.java` file in your `assignment5.zip` file.

### 5: Chaser Player [20 marks]

Create a class called `Chaser` that extends the `Player` class.

Your Chaser player will chase after a player on the opposing team. We will use a `RandomWalker` player as the opposing player when we test your code. If your player catches up to the `RandomWalker` player, your player should continue to follow it.

If you did not get your `RandomWalker` player working to test this, just use the `DummyPlayer` class that is given. Test your code with only two players in the game: your Chaser player and one opponent.

Mark breakdown: 20 marks for correctness

Put your `Chaser.java` file in your `assignment5.zip` file.

Note: When testing your code for your Chaser player, try adjusting the speed of your player. When your player moves faster it should catch the `RandomWalker` player efficiently. When your player moves slower you should be able to see that it is (usually) walking in a curve rather than a straight line to try and chase the `RandomWalker` player.

## 6: Catcher Player [20 marks]

Create a class called `Catcher` that extends the `Player` class.

The goal of your Catcher player is to try and catch opposing players on the field. Your player should not chase after players on its own team. Your player should stop chasing after a player opponent if they catch them (the `catchOpponent` method lets you know when you catch an opponent). If your player catches an opponent, it should continue to look for and chase after different opponents if there are any.

Mark breakdown: 20 marks for correctness

Put your `Catcher.java` file in your `assignment5.zip` file.

## Submission Recap

A complete assignment will consist of a single file (`assignment5.zip`) with the five files: `Stopping.java`, `Seeker.java`, `RandomWalker.java`, `Chaser.java`, and `Catcher.java`.