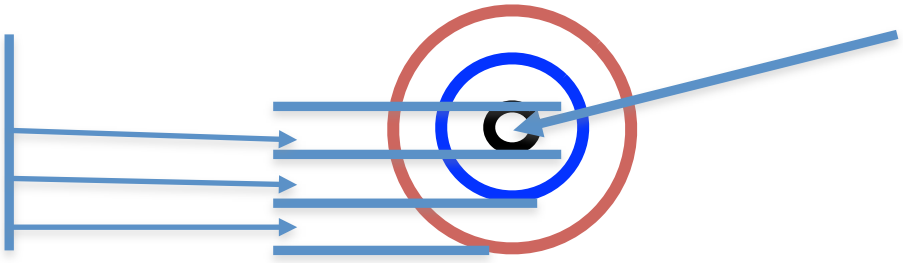
CISC 110 Lab 1

Work on this lab during the lab time or on your own time. Be sure you understand everything well, in order to prepare for the course tests and final project.

In this lab, you will start to write code that could be used as part of a shooter game, in which a player is being flown around in a plane and must shoot a stationary bull's eye target.

The target is composed of three concentric rings: an outer ring, a middle ring, and the inner circle (Bull-s eye). The diameter of the inner circle and the width of both the middle ring and outer ring is 20 pixels. The center point of the target is at (x, y) position (100, 150).



You will define the required constants and variables and set their initial values for the start of the game. In Lab 2, you will write the code that needs to be executed whenever the player shoots at the target. For now, you will need to *imagine* how this would work within the game. Later in the course you'll understand how the game could have been fully implemented.

In the game, there is a single player, their score (initially 0), two levels, and the speed of the plane they're in (which increases when they reach the second level).

When creating variables and constants, remember to use meaningful names and:

- Follow the syntax rules for creating variable names: names start with a letter, etc.
- Use the following common convention for variable names. Start with a lower-case letter and then capitalize the start of each word, with no underscores or spaces between words, for instance: planeSpeed
- Use the following convention for constant names. Use all upper-case letters and separate words with underscore characters, for instance BULLS_EYE.
- Don't use "keywords" as your names; keywords are reserved words used to perform specific tasks in ActionScript, for instance "var" to declare a variable. If you use variables that are two words long, you won't overlap with keywords. There is a list of keywords in Flash Help. Also, keywords are colored blue in the Flash editor.

Here is what you need to do:

1. Define the following constants:
   - Three constants to specify the scores the player will obtain for hitting a bull's eye (50), the middle ring (30), and the outer ring (10). For instance the constant for hitting the bulls eye could be called `BULLS_EYE` and must be set to 50.

   - A constant to specify the position of the X-coordinate and a constant to specify the position of the Y-coordinate of the center point of the target, which is (100, 150).

2. Add definitions for the following variables (decide which is the best data type for each of Number, int, uint, String, or Boolean):
    - A variable to store the name of the player; set it to a name of your choice
    - A variable to store the score; initially set it to 0
    - A variable to store the speed; initially set it to 5.5
    - A variable to store whether or not the player is at level 1; initially set it to true
    - A variable to store whether or not the player is at level 2; initially set it to false
    - Two variables to store the X-coordinate and Y-coordinate of the player's last shot.

3. Add assignment statements that set the variables storing the X-coordinate and Y-coordinate of the player's last shot to values that would make the shot a bull's eye, just for testing purposes. In the real game, those variables would be assigned by the player's actual shot.

4. Display the variables storing the player's last shot, using a trace statement. Display those variables with a label: "Last shot". Your output might look like this:

```
Last shot: (105,142)
```

5. Since the player has hypothetically hit the bull's eye, increase the player's score by the constant BULLS_EYE. Use a general assignment statement that would work even if the score was now 20 or 178 and even if we later decide to change what a bull's eye is worth. Don't just say: `score = 50;` You need to ***add BULLS_EYE*** to the current score, whatever that score is.

6. Assume that the player has reached the next level with their bull's eye. So set level 1 to false and level 2 to true and double the speed (since at a higher level the plane goes faster making it more difficult to shoot the target).

7. Display on the screen the player's new score and the values of the level variables and current speed, along with the position of their last shot. Write statements that would work correctly no matter what the values of the variables are.
   Your output might look like this:

   Last shot: (105, 142)
   Score: 50
   Level 1: false
   Level 2: true
   Speed: 11.0

8. Think of other things you might want in such a game to enhance the game play. Create variables and constants to keep track of them. Give them example values and display them with labeled output. Include comments to explain what their purpose is. Add at least 50% more code than you have already written with steps 1 to 7, so this step should be at least 1/3 of your total lines of code.