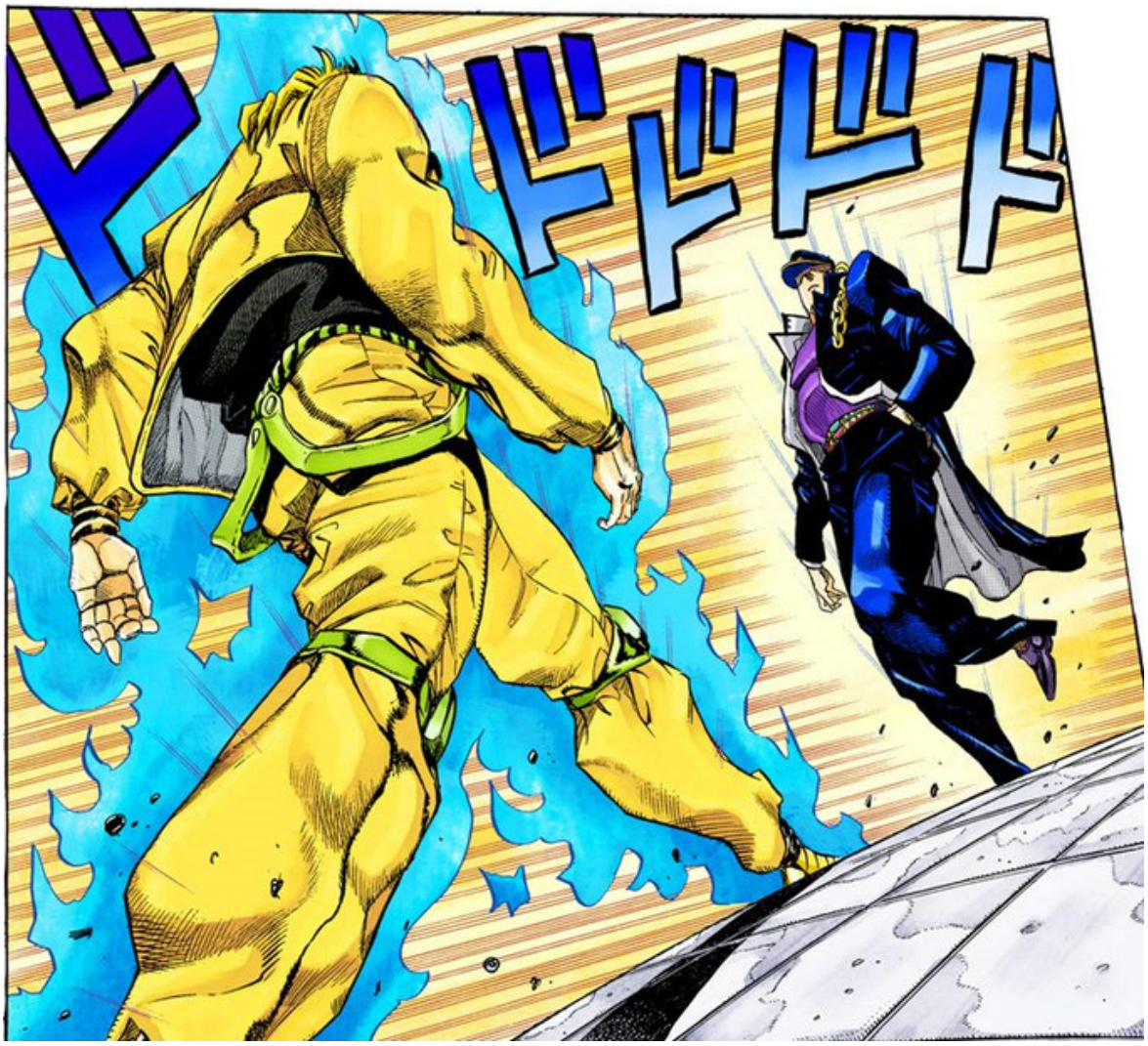
CS1580 - LAB12: Vs - Sir Rig's-A-Lot

Objective:

- Inheritance
- Polymorphism



Scenario:

You have finally destroyed the carnival, and have gotten the attention of the great Sir-Rig's-A-Lot himself. He is not too

pleased with your actions, and is seeking revenge. There is only one thing for your group to do now: beat him to death.

Requirements:

Much of the implementation of your classes will be up to you. Here are the requirements:

- You must use a base class to represent a **Person**
- That class must have 4 derived classes: **Puncher**, **Kicker**, **Insulter**, and **Boss**.
- Each class must have an attack value, a health value, and a speed value, with a random value in the ranges listed below (inclusive): (Set these values in a default constructor)

	Puncher	**Kicker**	**Insulter**	**Boss**
Attack	5-20	20-40	1-5	5-15
Health	70-100	30-50	100-300	200-450
Speed	0-100	0-100	0-100	0-100

- The puncher, kicker, and insulter class must have a function **attack**, which takes one object of type **person** as a parameter. This function should remove health from the parameter person equal to the attack of the calling object.
- The **boss** class must have a function **attack**, which takes an array of 3 **person**'s as a parameter. It should remove health from all people in the parameter array equal to the bosses attack.
- Each class must have a "print" function, which outputs the classes name, attack, health, and speed.
- Include whatever other functionality in your classes you may need.

If your **main** function:

- Set the random seed to 2

- Create a boss, puncher, kicker, and insulter.
- Print the information for all 4 of these people.
- Write a function that takes the boss and an array of 3 **Person**'s as parameters, which will be your battle simulator.
- Have the boss fight the other 3 people in turn based combat.
    - Turn order is based on speed. Higher speed people get to attack first. If there is a tie, break it in the order: boss first, puncher second, kicker third, insulter fourth
    - If anyone runs out of health, they can no longer fight.
    - The boss wins if all 3 people run out of health, the other fighters win if the boss runs out of health.
- Output a message stating the winner of the battle.

Submission / compilation:

As we are still working with multiple files, compile your code with *fg++ *.cpp -o [executable file name].* (Leave out the square brackets.)

As usual, submit your lab assignments through git.

- Navigate to your repository, and clone with https
- In putty/linux, *git clone (the link you just copied.)*
- **cd into the repository you just cloned**
- Write the code
- *git add .*
- *git commit -m "I'll miss you guys"*
- *git push*

Example:

---It's time to D-D-D-D-FIGHT!---

Boss:
Attack: 32
Health: 306
Speed: 91
Puncher:
Attack: 7
Health: 73
Speed: 62

Kicker:
Attack : 38
Health: 31
Speed: 40

Insulter:
Attack: 5
Health: 298
Speed: 85

--Commencing battle--

Sir Rigs-A-Lot wins!

Boss:
Attack: 32
Health: 247
Speed: 91

Some notes:

- You may run into problems if you don't use **virtual** functions in this assignment, as we are using derived types in an array of their base type objects.

BONUS! (+5):

Sending 3 people to fight Sir Rigs-A-Lot himself is... risky. Edit your code to work with any sized array of people. You can use either dynamic pointers or vectors for this.