Use the [GridWorld Quick Reference](https://secure-media.collegeboard.org/apc/ap_comp_sci_a_quick_reference.pdf) if needed.Test the code out during and after each step to make sure it is error free

**Part 0:**

Create a new Class named TeleportActorWorld w/ a main method

In the main method create a new world that is NOT 10x10, use the code below.

Code to use in the main:

ActorWorld world = new ActorWorld( new BoundedGrid<Actor>( #rows,#cols) );

// add actors to the world here

world.show();

**\*\* Test part 0 out before moving on to part 1 \*\***

**Part 1:**

Create a new Class named TeleportActor, make it extend Actor

Go to the TeleportActorWorld and add a default TeleportActor to the location (0,0)

New Classes/Methods to use:

extends Actor

world.add( Location loc, Actor actor )

new Location( int row, int col )

new TeleportActor( )

**\*\* Test Part 1 out before moving on to part 2 \*\***

**Part 2:**

Override the act method from the Actor class in the TeleportActor class.

Make the TeleportActor move to a random location on the Grid.

New Classes/Methods to use:

public void act()

new Location( int row, int col )

moveTo( Location newLocation )

getGrid().getNumCols()

getGrid().getNumRows()

Math.random()

**\*\* Test Part 2 out before moving on to part 3 \*\***

**Part 3:**

Each time the TeleportActor “teleports”, make it change to a random color.

New Classes/Methods to use:

new Color( int red, int green, int blue)

Math.random()

setColor( Color newColor )

**\*\* Test part 3 out before moving on to part 4 \*\***

**Part 4:**

Make the TeleportActor leave random colored Rocks in the locations that it has been.

*Hint:* Store the location before moving to a new location, then put a Rock in the old location after you move

New Classes/Methods to use:

getLocation()

new Rock( Color color )

putSelfInGrid( Grid<Actor> grid, Location loc )

**\*\* Test part 4 out before moving on to part 5 \*\***

**Part 5:**

Create an ArrayList to store the Locations that the TeleportActor has been to and do not allow it to go back to the locations that it has visited.

Test if the list contains the random location, if so pick another random location, otherwise move to that location.

*Hint:* Make an instance variable. If you make an instance variable what else do you need to make??

New Classes/Methods to use:

new ArrayList< Location >()

ArrayList.contains( )

**\*\* Test part 5 out before turning it in \*\***