Adam Cox

**2.**

ArrayList <Int> evenNumbers = new ArrayList<>();

**3.**

for ( i = 0; i <= 9; i++)

{

evenNumbers.add(i\*2);

}

**4.**

ArrayList<Car> cars = new ArrayList<Car>();

cars.add(new Car("Mustang", 2006, "tiger-striped"));

cars.add(new Car("MiniCooper", 2006, "lime green"));

**5.**

For (String address : addressList)

{

address.display();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* HGdriver.Java

\* <Adam Cox>

\*

\* This gets rid of all but one participant and claims them winner

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** java.util.\*;

**public** **class** HungerGames

{

**public** **static** **void** main(String[] args)

{

ArrayList<String> tributes = **new** ArrayList<>();

tributes.add("Cato");

tributes.add("Katniss");

tributes.add("Peeta");

tributes.add("Rue");

HungerGames.*play*(tributes);

} // end main

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//This gets rid of the other tributes

**public** **static** **void** play(ArrayList<String> tributes)

{

**for**(**int** i = 0; i <= tributes.size(); i++) //loop to remove tributes

{

**int** z= (**int**) (tributes.size() \* Math.*random*()); //Random int

tributes.remove(z);

} //end for loop

System.***out***.print("This game is over, and the victor is" +tributes.get(0));

}//end of play

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* ElevatorDriver.Java

\* <Adam Cox>

\*

\* This is the driver for Elevator

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** java.util.\*;

**public** **class** ElevatorDriver

{

**public** **static** **void** main(String[] args)

{

Scanner stdIn = **new** Scanner(System.***in***);

**int** selection; // Menu selection

**double** weight; // Weight of passenger

**boolean** addPassengers = **true**; // allowed to add passengers

Elevator elevatorOne = **new** Elevator(1000.0);

System.***out***.println("Elevator has a " +

Math.*round*(elevatorOne.MAX\_LOAD) + " pound load limit.\n");

**while** (addPassengers)

{

**while** (elevatorOne.isOverWeightLimit())

{

elevatorOne.display();

System.***out***.print("Elevator is overloaded. " +

"Who will take the stairs? ");

selection = stdIn.nextInt();

elevatorOne.removePassenger(selection);

System.***out***.println();

} // end while

System.***out***.print("Input a weight in pounds of passenger" +

" (0 to finish): ");

weight = stdIn.nextDouble();

**if** (weight == 0)

{

addPassengers = **false**;

}

**else**

{

elevatorOne.addPassenger(weight);

System.***out***.println();

}

} // end while

elevatorOne.display();

System.***out***.println();

elevatorOne.everyoneOut();

elevatorOne.display();

} // end main

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Elevator.Java

\* <Adam Cox>

\*

\* This program adjust the weights of passengers and gives an alert.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** java.util.\*;

**public** **class** Elevator

{

**public** **final** **double** MAX\_LOAD; //This set the max weight

**public** **double** combinedWeight = 0; //This is the passenger weight

**public** ArrayList<Double> passengerWeights = **new** ArrayList<>(); //Each Passenger Weight

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** Elevator(**double** weight)

{

MAX\_LOAD = weight;

} // end constructor

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// This determines if the elevator is over the weight limit.

**public** **boolean** isOverWeightLimit()

{

**return** (combinedWeight > MAX\_LOAD);

} // end isOverWeightLimit

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// This adds a passenger weight to the arraylist and combined

**public** **void** addPassenger(**double** body)

{

System.***out***.print("A "+ body + " passenger entered the elevator.\n");

passengerWeights.add(body);

combinedWeight += body;

**if** (combinedWeight > MAX\_LOAD)

{

System.***out***.print("Warning! The elevator has surpassed its maximum "

+ "weight capacity of "+ MAX\_LOAD +" pounds.\n");

}

} // end addPassenger

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// This removes the passenger selected

**public** **void** removePassenger(**int** person)

{

**int** personIndex = person - 1; // Subtract 1 from user input

**if** (person >=1 && person <=passengerWeights.size())

{

combinedWeight -= passengerWeights.get(personIndex);

System.***out***.print("A "+ passengerWeights.get(personIndex) + " pound passenger"

+ " has left the elevator.\n");

passengerWeights.remove(personIndex);

}

**else**

{

System.***out***.print("Error! Invalid selection.\n");

}

} // end removePassenger

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Removes everyone from the Elevator.

**public** **void** everyoneOut()

{

passengerWeights.clear();

combinedWeight = 0;

System.***out***.print("All the passengers have exited.\n");

} // end everyoneOut

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Displays the weight of each person

**public** **void** display()

{

**if** (combinedWeight == 0)

{

System.***out***.print("There are currently no passengers left on the elevator.");

}

**else**

{

System.***out***.print("Here are the passenger weights:\n");

**for** (**int** i = 0; i< passengerWeights.size(); i++) //displays weights

{

System.***out***.printf(i+1 + ") " +passengerWeights.get(i)+ "\n");

}

}

} // end display

} // end elevator

Elevator has a 1000 pound load limit.

Input a weight in pounds of passenger (0 to finish): 200

A 200.0 passenger entered the elevator.

Input a weight in pounds of passenger (0 to finish): 100

A 100.0 passenger entered the elevator.

Input a weight in pounds of passenger (0 to finish): 300

A 300.0 passenger entered the elevator.

Input a weight in pounds of passenger (0 to finish): 200

A 200.0 passenger entered the elevator.

Input a weight in pounds of passenger (0 to finish): 400

A 400.0 passenger entered the elevator.

Warning! The elevator has surpassed its maximum weight capacity of 1000.0 pounds.

Here are the passenger weights:

1) 200.0

2) 100.0

3) 300.0

4) 200.0

5) 400.0

Elevator is overloaded. Who will take the stairs? 6

Error! Invalid selection.

Here are the passenger weights:

1) 200.0

2) 100.0

3) 300.0

4) 200.0

5) 400.0

Elevator is overloaded. Who will take the stairs? 2

A 100.0 pound passenger has left the elevator.

Here are the passenger weights:

1) 200.0

2) 300.0

3) 200.0

4) 400.0

Elevator is overloaded. Who will take the stairs? 4

A 400.0 pound passenger has left the elevator.

Input a weight in pounds of passenger (0 to finish): 0

Here are the passenger weights:

1) 200.0

2) 300.0

3) 200.0

All the passengers have exited.

There are currently no passengers left on the elevator.