

Welder, trapezemike from <https://pixabay.com/en/welder-welding-work-labor-job-car-3018425/>, CC0 Public Domain

A local manufacturing company has final products delivered by lift truck from the preparation department. As the lift trucker picks the item up, the product is weighed and a shipping ticket is printed and the driver attaches the ticket to the product. The product is measured in pounds with a unique number specifying the various department specifications. Once the item is placed in queue, the computer determines the shipping order. The shipping order is the heaviest items are moved to the highest priority due to transportation regulations.

The ShippingTicket class is used to represent the information on the product ticket and the ShippingQueue class is used to keep track of the products that are waiting to be shipped.

public class ShippingTicket

{

/\* instance variables not shown \*/

public ShippingTicket(double weight, String id)

{

/\* implementation not shown \*/)

}

public double getWeight()

{

/\* implementation not shown \*/)

}

public String getID()

{

/\* implementation not shown \*/)

}

}

public class ShippingQueue

{

private ArrayList<ShippingTicket> products;

//Constructs a ShippingQueue with an empty products ArrayList

public ShippingQueue()

{

code for part a

}

public ArrayList<ShippingTicket> getProducts()

{

return products;

}

// Adds a ShippingTicket to the end of the products ArrayLists

public void addProduct(double weight, String id)

{

code for part b

}

/\*\* Returns the number of products waiting to be shipped

\* That exceed the given weight

\*/

public int productsExceeding(double weight)

{

code for part c

}

/\*\* Returns the heaviest product. If there are no products

\* Waiting to be shipped, returns null.

\*/

public ShippingTicket nextProduct()

{

code for part d

}

Instructions: You may print and hand write your answers to the following questions. This will be how you will be completing this part on the actual AP Exam. To submit if you hand write, you will need to take an image or scan and submit. For this practice, you may also enter your answers directly into this document, save and submit

Part a: The constructor for ShippingQueue must assign to the variable products an empty ArrayList. Complete the following definition

//Constructs a ShippingQueue with an empty products ArrayList

public ShippingQueue()

{

products = new ArrayList<ShippingTickets>();

}

Part b: Complete the following definition for addProduct:

// Adds a ShippingTicket to the end of the products ArrayList

public void addProduct(double weight, String id)

{

ShippingTicket newTicket = new ShippingTicket(weight, id);

products.add(newTicket);

}

Part c: Complete the following definition for products Exceeding:

/\*\* Returns the number of products waiting to be shipped

\* That exceed the given weight

\*/

Public int productsExceeding(double weight)

{

int numberOver = 0;

for (int count = 0; count < products.size() - 1; count++)

{

if (products.get(count).getWeight > weight)

{

numberOver++;

}

}

return numberOver;

}

Part d: Complete the following definition for nextProduct

/\*\* Returns the heaviest product. If there are no products

\*Waiting to be shipped, return null.

\*/

public ShippingTicket nextProduct()

{

int heaviestIndex = 0;

int heaviestWeight = 0;

if (products.size() = 0)

{

return null;

}

for (int count = 0; count < products.size() - 1; count++)

{

if (products.get(count).getWeight > heaviestWeight)

{

heaviestIndex = count;

heaviestWeight = products.get(count).getWeight;

}

}

return products.get(heaviestIndex);

}