# CS152 Spring 2017

# Project#1 Packets

Packet objects have unique ID number, weight in pounds with 2 decimals, and state abbreviations for their destination and origin.

Class Packet describes one packet and has variables idNumber, weight, and stateDest and stateOrig of type int, double, String and String respectively. In addition, it has the following methods

* boolean isHeavy () that returns true when packet is above 10 pounds, and false otherwise.
* boolean isInState() that returns true when packet is shipped to the same destination state as is the state of origin.
* String toString() returns String which is one line string representation of Packet objects.
* double getWeight() returns packet weight.

Create an input file called "packetData.txt" with the following 7 lines. Each line in the "packetData.txt" file has information about one packet object.

1001 8.37 CA MO

1002 2.17 CT CA

1003 11.35 NY NY

1004 3.77 MA MA

1005 9.99 FL CT

1006 14.91 VT MA

1007 4.97 TX CA

Class Packages has shipmentList and totalWeight variables. Variable called shipmentList contains the collection of all packets. Variable shipmentList is of ArrayList type. All objects in the ArrayList are of Packet type. Totalweight variable specifies the total weight of all packets. The constructor assigns variables shipmentList and totalWeight. Variable totalWeight is initialized to 0.0 when declared in the class and updated when each packet is read from the file.. Variable shipmentList is assigned by reading the data from the input file specified above. Each line in the file has data about one packet object. As you read the data from one line in the input file a Packet object should be added to the shipmentList Class Packages has methods:

* String toString() which returns String representation about entire collection of packets with one packet object specified per line.
* void displayHeavyPackages() which displays all packets that are heavy .
* void displayInStatePackages() which displays all packets with the same origin and destination state. Your code must use method isInState().
* Packet MaxWeightPacket() returns heaviest packet object.
* void displayAverageWeight () displays average weight (with two decimals) of all packets. Use full sentence.

All of the methods are written by using variables shipmentList and totalWeight.

Your application should also have class TestPackages with only main method in it, in addition to classes Packet and Packages.

EXTRA CREDIT (1 point)

Add method String destinationLocation() to Packet class. Method returns String "Local" if destination state is one of CT, MA, RI, or NY, and returns String "NonLocal" for all other states.

Add methods to Packages class:

* void displayLocalPackets() which displays one empty line, then title "LOCAL PACKETS", and next it displays all local packets that are shipped to any local state (Display one packet per line). Must use method destinationLocation in the code.
* void displayNonLocalPackets() which displays all packets that are not shipped to a local destination.

Must use method destinationLocation to write the code.

PROGRAM RUN outline:

|  |
| --- |
| ALL PACKETS  Display all packets by calling toString() method from class Packages.  All HEAVY PACKETS  Display all heavy packets  All IN-STATE PACKETS  Display all packets that are shipped to the same origin and destination state.  The packet object with max weight is: X X X X.  The average weight of all packets is XXX.  EXTRA CREDIT: LOCAL PACKETS  Display all local packets  EXTRA CREDIT: NONLOCAL PACKETS  Display all packets that are not local |

**SUBMIT one word or PDF document** named P1your-last-name. Put Your name, class section, project number, and date of your submission in the upper left corner. Document should have picture of UML diagram, code for each of the classes, and picture of BlueJ output window. Do not submit BlueJ files, and do not ZIP your submission. Read document about style requirements to make sure that your style is as required. Style is graded too.

|  |  |  |
| --- | --- | --- |
| SUBMIT: | POSSIBLE POINTS | STUDENT GRADE |
| UML Design | 1 point |  |
| Code for each of the three classes. | 8 points |  |
| Program run documented in a Word file. | 1 point |  |
| Extra Credit | 1 point |  |

**CODE**: Highlight the code in BlueJ window, press CTRL and c together to put the code into the paste buffer. Next click the place in your word document where you want the code to be placed and press CTRL and v to paste it there.

**PICTURE OF PROGRAM RUN WINDOW**: Run your program in BlueJ. Resize the window so that displays only output info. While output window is active press ALT and PRTSCRN together to put picture of the entire active window into paste buffer. Click the place in your word document where you want it and press press CTRL and v to paste it there.

**PICTURE OF UML:** When you finish drawing the picture of UML ( in Paint or any drawing software tool) press ALT and PRTSCRN together to put picture of the entire active window into paste buffer. Click the place in your word document where you want it and press CTRL and v to paste it there.