

CMSC131, Spring 2020, Quiz #4 (Memory Map)

Deadline: Friday, April 24, 2:00 pm (No late deadline)

Specifications

Draw a memory map for the code you see below, until the execution reaches the point indicated by the comment `/* HERE */`. In your diagram:

- You must have a stack, heap, and static memory sections as illustrated by the examples at:

<http://www.cs.umd.edu/~nelson/classes/resources/MemoryMapsInformation/MemoryMapsInformation.pdf>

- Identify each frame as illustrated by the previous examples.
- Draw your variables as they are encountered during program execution.

```
public class Passenger {
    private String name;
    private int ticketCost;
    private StringBuffer luggage;
    private static final int LUGGAGE_COST = 20;

    public Passenger(String name, int ticketCost) {
        this.name = name;
        this.ticketCost = ticketCost;
        luggage = new StringBuffer();
    }

    public Passenger addLuggage(String desc) {
        ticketCost += LUGGAGE_COST;
        luggage.append(desc);

        return this;
    }

    public int getTicketCost() {
        return ticketCost;
    }

    public Passenger reduceCost(int by) {
        ticketCost -= by;

        /* HERE */
        return this;
    }

    public String toString() {
        return "Passenger [name=" + name + ", ticketCost=" + ticketCost + ", luggage=" + luggage + "];"
    }
}

public class Driver {

    public static void inspect(Passenger[] allp, int which) {
        int cost = allp[which].getTicketCost();
        allp[which].reduceCost(cost / 2);

        System.out.println(allp[1]);
        System.out.println(allp[2]);
    }

    public static void main(String[] args) {
        Passenger laura = new Passenger("Laura", 300);
        laura.addLuggage("guitar");
        Passenger tom = new Passenger("Tom", 125);
        tom.addLuggage("mac").addLuggage("PC");

        Passenger[] friends = new Passenger[4];
        friends[1] = laura;
        friends[2] = tom;

        inspect(friends, 1);
    }
}
```