CS 361

Combat Repair Yard Simulation

You are to implement a time stepping simulation of a World War II combat vehicle repair yard.

1. Class Vehicle
   1. Id num
   2. Type V1…V5
   3. List of usable parts. P1…Pn (n = type# + 3) so a type V1 has 4 parts max.

|  |  |
| --- | --- |
| Vehicle type | Part list |
| 1 | 1, 2, 3, 4 |
| 2 | 2, 4, 6, 8, 10 |
| 3 | 3, 5, 7, 9, 11,13 |
| 4 | 10, 11, 12, 13, 14, 15, 16 |
| 5 | 1, 2, 3, 5, 7, 11, 13, 17 |

Generate a number of damaged vehicles, all such vehicles have at least 1 good part and at least one failed part every day. The number of vehicles arriving will be average = 3, standard div = 1

1. Create a Yard Master function which will score the vehicles as repairable or not. If there are at least 50% good parts then the vehicle will go to the repair queue, else scrap yard.
2. From the repair queue, damaged vehicles will be assigned to one of three repair bays on a first available basis.
3. Create a class repair bay, each bay has a vehicle pointer, vptr.
4. Create an array of 3 repair bays
5. Repair times = sum of the damaged part numbers
6. Replacement parts come first from the scrap yard if available.
7. IF the part is NOT available then warehouse.
8. The warehouse has a list of parts, 75% chance of any one particular part, 50% of having multiple copies.
9. IF the warehouse does not have the required part, there is a time delay to order one, 10% chance every day that a needed part will arrive. If 5 days pass without arriving the vehicle will removed from the repair bay and placed in the scrap yard.
10. Once repaired, vehicles will be tested, each part will have a 95% chance of passing tests, if fail then the vehicles go back to the repair bay.
11. Each time step print to screen the state of the Repair Yard, e.g. a log , of what is in each data structure.
12. Each day is divided into 3 time steps. User prompt for each additional day.

Design of software:

Objects, Functions & Data Structures

1. Class Repair Yard
   1. An array of 3 repair bays
      1. Vehicle pointer
      2. Status {avail, in\_use}
      3. Est time to complete
   2. A list– arrived
   3. A priority queue – to\_be\_repaired
   4. A scrap yard – a list of damaged vehicles
   5. Warehouse – a vector of available parts
   6. Yardmaster function that scores vehicles
2. Class Vehicle (see above)
3. Damaged Vehicle Creator Function