Cheeto Bag

**Use case:** This is a Cheeto bag class that simulates a real bag of Cheetos and the state of the said Cheeto bag. Functionally, this class should be able to tell the number of Cheetos in the bag, the size of the bag, if the bag is full, and what flavor of Cheetos the bag contains. This could be used in a video game or for testing for a product

**Name: Cheeto Bag**

**Data members:**

int numCheetos – the number of Cheetos currently in the bag

bool isFull – is the bag full or not

bool isOpened – is the bag opened or still sealed

string flavor – the flavor of the Cheetos inside the bag

string size – the size of the bag, correlates to numCheetos

float staleness - the staleness of the Cheetos in the bag

**Methods:**

CheetoBag(string bagSize, string cheetoFlavor) – constructor to make a Cheeto bag of a certain size with a certain flavor

Interfaces:

int getNumCheetos() – gets the number of Cheetos in the bag

bool isBagFull() – returns true or false depending on if the bag is full.

bool isBagOpen() – returns true or false depending on if the bag was opened

string getFlavor() – returns flavor of the Cheetos in the bag

string getSize() -returns the size of the Cheeto bag

void openBag() - opens the Cheeto bag

void closeBag() - "closes" the bag so the Cheetos don't go stale.

int takeHandful() - gets a handful of a random, realistic number of Cheetos and takes them out of the bag

void eatHandful() - removes the handful of Cheetos from physical existence.

int shareCheetos(int numToShare) - gives Cheetos away so someone else can eat them, returns cheetos left in the bag

float goStale(int timeOpen) - if the bag is opened for timeOpen amount of time (in days), the Cheetos go stale a certain percent, with 1 being the not stale and 0 being fully stale. returns how stale they are.

void print() - prints the current information about the Cheeto bag

**UML Diagram:**

|  |
| --- |
| Cheeto Bag |
| -int numCheetos  -bool isFull  -bool isOpened  -string flavor  -string size  -float staleness |
| +CheetoBag(string bagSize, string cheetoFlavor)  +void openBag()  +void closeBag()  +int takeHandful()  +void eatHandful()  +int shareCheetos(int numToShare)  +float goStale(int timeOpen)  +void print() |

Laptop

**Use case:** The laptop class can be used in a video game or in a simulation of an event. This class would simulate a basic laptop, having a charge, a brightness, and a workload. Its battery charge would reduce based on the workload and brightness.

**Name: Cheeto Bag**

**Data Members:**

int battery - percent of charge left in the battery as a percentage from 0 to 100

int brightness - the brightness of the computer, in a range from 1 to 10

int workload - the amount of work the laptop is exerting on a scale of 0 to 100

**Methods:**

Laptop(int newBattery, int newBrightness) - makes a new laptop with a battery and brightness

Interfaces:

void getCharge() - returns the amount of charge the laptop has

void getBrightness() - returns the brightness of the laptop

void getWorkload() - returns the workload of the laptop

void setBrightness(int newBright) - sets the brightness to newBright, a number from 1 to 10

void setWorkload(int newWorkload) - sets the workload to newWorkload, a number from 0 to 100

int chargeBattery() - charges the battery to 100 percent

int chargeBattery(int chargePercent) - charges the battery of the laptop for chargePercent amount

int decreaseBattery(int percent) - decreases battery a certain percent

int decreaseBattery(int totalWork, int currBrightness, int time) - decreases the battery based on total work and current brightness over a given period of time

int autoBrightness(int batteryLeft, int totalWorkload) - sets the brightness based on amount of battery left and the workload, returns the new brightness level on a scale of 1 to 10

void print() - prints the current information about the laptop

**UML Diagram:**

|  |
| --- |
| Laptop |
| -int battery  -int brightness  -int workload |
| +Laptop(int newBattery, int newBrightness)  +int chargeBattery()  +int chargeBattery(int chargePercent)  +int decrease battery(int percent)  +int decreaseBattery(int totalWork, int currBrightness, int time)  +int autoBrightness(int batteryLeft, int totalWorkload)  +void print() |