NAME : JAGADEESH R

REGNO : 2021506314

ADS Lab06 :

USECASE : STATIONERY MANAGEMENT

PROGRAM : EXCEPTION HANDLING

SOURCE CODE :

```
#include <iostream>
#include <vector>
#include <string>
class Stationery {
public:
   std::string name;
   int quantity;
   Stationery(const std::string& name, int quantity) : name(name), quantity(quantity) {}
   void decreaseQuantity(int amount) {
      if (amount <= 0) {
         throw std::invalid_argument("Invalid quantity to decrease.");
      }
      if (amount > quantity) {
         throw std::runtime_error("Insufficient quantity.");
      }
      quantity -= amount;
   }
};

int main() {
   std::vector<Stationery> stock;
   stock.push_back(Stationery("Pen", 10));
   stock.push_back(Stationery("Notebook", 5));
   stock.push_back(Stationery("Eraser", 3));
```

```
    try {
        std::cout << "Initial stock:" << std::endl;
        for (const auto& item : stock) {
            std::cout << item.name << ": " << item.quantity << std::endl;
        }
        stock[0].decreaseQuantity(5);
        std::cout << "\nAfter decreasing Pen quantity:" << std::endl;
        for (const auto& item : stock) {
            std::cout << item.name << ": " << item.quantity << std::endl;
        }
        stock[1].decreaseQuantity(-2);
    }
    catch (const std::invalid_argument& e) {
        std::cerr << "Error: " << e.what() << std::endl;
    }
    catch (const std::runtime_error& e) {
        std::cerr << "Error: " << e.what() << std::endl;
    }

    return 0;
}
```

OUTPUT :

Output

```
/tmp/Tl9tMMbYht.o
Initial stock:
Pen: 10
Notebook: 5
Eraser: 3

After decreasing Pen quantity:
Pen: 5
Notebook: 5
Eraser: 3
ERROR!
Error: Invalid quantity to decrease.
```