

Assignment 3.1: Simple Vending Machine EASY Memory Limit: 200000 KB Time Limit: 1000 ms

Simple Vending Machine

Due date: Thu, 29 Aug 2024 12:00 AM
Created date: Thu, 22 Aug 2024 11:03 PM

Problem Submission (3)

Assignment 3.1: Simple Vending Machine

Problem

You are tasked with developing a simple program for a vending machine. The machine dispenses drinks based on user input and the amount of money inserted. The vending machine offers three drinks:

- 1: Water (7 Baht)
- 2: Soda (13 Baht)
- 3: Juice (20 Baht)

Your job is to implement the following functionality:

- The user selects a drink by entering a number (1, 2, or 3).
- The user then inserts an amount of money.
- The machine checks if the inserted amount is sufficient to buy the selected drink.

```
1 // Remove this
2 // @formatter:off
3 #include <iostream>
4
5 int main(void) {
6     // Variables to store input
7     int ITEM;
8     float FUND;
9
10    // Get item number (check if it is 1, 2, or 3)
11    while (true) {
12        std::cout << "ITEM: ";
13        int ITEM;
14        if (std::cin.get() != '\n') {
15            continue;
16        }
17        if (ITEM < 1 || ITEM > 3) {
18            std::cout << "Invalid item!\n";
19            continue;
20        }
21        break;
22    }
23
24    // Get fund amount (check if it is 1, 2, or 3)
25    while (true) {
26        std::cout << "FUND: ";
27        float FUND;
28        if (std::cin.get() != '\n') {
29            continue;
30        }
31        if (FUND < 0) {
32            std::cout << "Invalid fund!\n";
33            continue;
34        }
35        break;
36    }
37
38    // Check if the fund is sufficient to buy the selected item
39    if (ITEM == 1 && FUND < 7) {
40        std::cout << "Insufficient fund! Water!\n";
41        return 0;
42    }
43    if (ITEM == 2 && FUND < 13) {
44        std::cout << "Insufficient fund! Soda!\n";
45        return 0;
46    }
47    if (ITEM == 3 && FUND < 20) {
48        std::cout << "Insufficient fund! Juice!\n";
49        return 0;
50    }
51
52    // Dispense the drink
53    if (ITEM == 1) {
54        std::cout << "Dispensing your drink: Water!\n";
55    }
56    if (ITEM == 2) {
57        std::cout << "Dispensing your drink: Soda!\n";
58    }
59    if (ITEM == 3) {
60        std::cout << "Dispensing your drink: Juice!\n";
61    }
62
63    // Print the change
64    std::cout << "Change: " << FUND - (ITEM == 1 ? 7 : (ITEM == 2 ? 13 : 20)) << "\n";
65
66    return 0;
67 }
```

C

Submit

Assignment 3.2: Complex Pricing System with Discounts and Tax MEDIUM Memory Limit: 200000 KB Time Limit: 1000 ms

Complex Pricing System with Discounts and Tax

Due date: Thu, 29 Aug 2024 12:00 AM
Created date: Fri, 23 Aug 2024 12:19 AM

Problem Submission (7)

Assignment 3.2: Complex Pricing System with Discounts and Tax

Problem

You are tasked with developing a pricing system that calculates the final price of a product after applying relevant discounts and taxes. The system needs to account for different types of products and promotions.

The system should follow these rules:

- Product Types:**
 - Type 1: Electronics
 - Type 2: Clothing
 - Type 3: General Goods
- Promotions:**
 - If a promotion is available ('has_promotion' = 1):
 - For Electronics and Clothing:

```
1 // Remove this
2 // @formatter:off
3 #include <iostream>
4
5 int main(void) {
6     // Variables to store input
7     int type;
8     float price;
9     int has_promotion;
10    int type_discount;
11    float discount;
12
13    // Get product type (check if it is 1, 2, or 3)
14    while (true) {
15        std::cout << "Type: ";
16        int type;
17        if (std::cin.get() != '\n') {
18            continue;
19        }
20        if (type < 1 || type > 3) {
21            std::cout << "Invalid type!\n";
22            continue;
23        }
24        break;
25    }
26
27    // Get price
28    while (true) {
29        std::cout << "Price: ";
30        float price;
31        if (std::cin.get() != '\n') {
32            continue;
33        }
34        if (price < 0) {
35            std::cout << "Invalid price!\n";
36            continue;
37        }
38        break;
39    }
40
41    // Get has_promotion
42    while (true) {
43        std::cout << "Has Promotion: ";
44        int has_promotion;
45        if (std::cin.get() != '\n') {
46            continue;
47        }
48        if (has_promotion < 0 || has_promotion > 1) {
49            std::cout << "Invalid has_promotion!\n";
50            continue;
51        }
52        break;
53    }
54
55    // Get type_discount
56    while (true) {
57        std::cout << "Type Discount: ";
58        int type_discount;
59        if (std::cin.get() != '\n') {
60            continue;
61        }
62        if (type_discount < 1 || type_discount > 3) {
63            std::cout << "Invalid type_discount!\n";
64            continue;
65        }
66        break;
67    }
68
69    // Calculate discount
70    float discount;
71    if (type_discount == 1) {
72        // For Electronics
73        if (type == 1) {
74            discount = 0.10 * price;
75        }
76        else if (type == 2) {
77            discount = 0.05 * price;
78        }
79        else {
80            discount = 0.00 * price;
81        }
82    }
83    else if (type_discount == 2) {
84        // For Clothing
85        if (type == 1) {
86            discount = 0.05 * price;
87        }
88        else if (type == 2) {
89            discount = 0.10 * price;
90        }
91        else {
92            discount = 0.00 * price;
93        }
94    }
95    else if (type_discount == 3) {
96        // For General Goods
97        if (type == 1) {
98            discount = 0.00 * price;
99        }
100       else if (type == 2) {
101           discount = 0.00 * price;
102       }
103       else {
104           discount = 0.00 * price;
105       }
106   }
107
108   // Calculate final price
109   float final_price = price - discount;
110
111   // Print final price
112   std::cout << "Final Price: " << final_price << "\n";
113
114   return 0;
115 }
```

C

Submit

Assignment 3.2: Complex Pricing System with Discounts and Tax MEDIUM Memory Limit: 200000 KB Time Limit: 1000 ms

Complex Pricing System with Discounts and Tax

Due date: Thu, 29 Aug 2024 12:00 AM
Created date: Fri, 23 Aug 2024 12:19 AM

Problem Submission (7)

Assignment 3.2: Complex Pricing System with Discounts and Tax

Problem

You are tasked with developing a pricing system that calculates the final price of a product after applying relevant discounts and taxes. The system needs to account for different types of products and promotions.

The system should follow these rules:

- Product Types:**
 - Type 1: Electronics
 - Type 2: Clothing
 - Type 3: General Goods
- Promotions:**
 - If a promotion is available ('has_promotion' = 1):
 - For Electronics and Clothing:

```
1 // Remove this
2 // @formatter:off
3 #include <iostream>
4
5 int main(void) {
6     // Variables to store input
7     int type;
8     float price;
9     int has_promotion;
10    int type_discount;
11    float discount;
12
13    // Get product type (check if it is 1, 2, or 3)
14    while (true) {
15        std::cout << "Type: ";
16        int type;
17        if (std::cin.get() != '\n') {
18            continue;
19        }
20        if (type < 1 || type > 3) {
21            std::cout << "Invalid type!\n";
22            continue;
23        }
24        break;
25    }
26
27    // Get price
28    while (true) {
29        std::cout << "Price: ";
30        float price;
31        if (std::cin.get() != '\n') {
32            continue;
33        }
34        if (price < 0) {
35            std::cout << "Invalid price!\n";
36            continue;
37        }
38        break;
39    }
40
41    // Get has_promotion
42    while (true) {
43        std::cout << "Has Promotion: ";
44        int has_promotion;
45        if (std::cin.get() != '\n') {
46            continue;
47        }
48        if (has_promotion < 0 || has_promotion > 1) {
49            std::cout << "Invalid has_promotion!\n";
50            continue;
51        }
52        break;
53    }
54
55    // Get type_discount
56    while (true) {
57        std::cout << "Type Discount: ";
58        int type_discount;
59        if (std::cin.get() != '\n') {
60            continue;
61        }
62        if (type_discount < 1 || type_discount > 3) {
63            std::cout << "Invalid type_discount!\n";
64            continue;
65        }
66        break;
67    }
68
69    // Calculate discount
70    float discount;
71    if (type_discount == 1) {
72        // For Electronics
73        if (type == 1) {
74            discount = 0.10 * price;
75        }
76        else if (type == 2) {
77            discount = 0.05 * price;
78        }
79        else {
80            discount = 0.00 * price;
81        }
82    }
83    else if (type_discount == 2) {
84        // For Clothing
85        if (type == 1) {
86            discount = 0.05 * price;
87        }
88        else if (type == 2) {
89            discount = 0.10 * price;
90        }
91        else {
92            discount = 0.00 * price;
93        }
94    }
95    else if (type_discount == 3) {
96        // For General Goods
97        if (type == 1) {
98            discount = 0.00 * price;
99        }
100       else if (type == 2) {
101           discount = 0.00 * price;
102       }
103       else {
104           discount = 0.00 * price;
105       }
106   }
107
108   // Calculate final price
109   float final_price = price - discount;
110
111   // Print final price
112   std::cout << "Final Price: " << final_price << "\n";
113
114   return 0;
115 }
```

C

Submit