Hospital Bill Calculator

Hissamuddin Shaikh

CSCI 201

23rd of June 2018

St. Cloud State University

**Hospital Bill Calculator**

**Problem Statement**:

Create a program that calculates and displays, the cost of a Patient undergoing a certain type of surgery, along with their pharmaceutical expenses and occupancy charges at the Hospital.

**Input/Output Description:**

Input: The Type of Surgery the Patient has gone through.

The Pharmaceutical Medication received by the Patient.

The Number of Days spent in Hospital care.

Output: The Number of Days spent at the Hospital.

The Hospital Charge for each Day spent.

The Total Cost of Surgery and Medications.

The Total Cost of Surgery, Medications and Hospital Stay.

**Variables:**

* ***Variables for the Surgery (surgeryType) class:***
  + *Arm*
    - A private variable constructed to store the Cost for Arm Surgery.
    - Type: int
  + *Leg*
    - A private variable constructed to store the Cost for Leg Surgery.
    - Type: int
  + *Eye*
    - A private variable constructed to store the Cost for Eye Surgery.
    - Type: int
  + *Brain*
    - A private variable constructed to store the Cost for Brain Surgery.
    - Type: int
  + *Heart*
    - A private variable constructed to store the Cost for Heart Surgery.
    - Type: int
* ***Variables for the Pharmacy (pharmacyType) class:***
  + *Med\_A*
    - A private variable constructed to store the Cost for Medicine A.
    - Type: int
  + *Med\_B*
    - A private variable constructed to store the Cost for Medicine B.
    - Type: int
  + *Med\_C*
    - A private variable constructed to store the Cost for Medicine C.
    - Type: int
  + *Med\_D*
    - A private variable constructed to store the Cost for Medicine D.
    - Type: int
  + *Med\_E*
    - A private variable constructed to store the Cost for Medicine E.
    - Type: int
* ***Variables for the Patient Account (patientAccountType) class:***
  + *Days*
    - A private variable constructed to store the Number of Days spent at the Hospital.
    - Type: int
  + *totalCharges*
    - A private variable constructed to store the Total Charge of the Surgery and the Medicines.
    - Type: int
  + *Rate*
    - A private variable constructed to store the Charge for a Single Day spent at the Hospital.
    - Type: int
  + *surgeryCharges*
    - Object to access the class *surgeryType*.
  + *pharmacyCharges*
    - Object to access the class *pharmacyType*.
  + *D*
    - A variable used in the parametrical constructor for the Class, and in the public function *setDays*, for the Number of Days spent at the Hospital.
    - Type: int
  + *Total*
    - A variable used in the parametrical constructor for the Class, and in the public function *setTotalCharges*, for the Total Cost of the Surgery and Medicine.
    - Type: int
  + *R*
    - A variable used in the parametrical constructor for the Class, and in the public function *setRate*, for the Total Cost of a Single Day spent at the Hospital.
    - Type: int
  + *Option*
    - A variable used in the public function *setPharmacyCharges* and *setSurgeryCharges* , to store the selection of the User as to what the Patient is to be charged for.
    - Type: char
* ***Variables for the Main Program (Function):***
  + *Choice*
    - A variable to store the selection of the User, as to what type of Surgery, as well as all the Medicines, the Patient is to be charged for.
    - Type: char
  + *Final\_Choice*
    - A variable to store the selection of the User, as to what type of Surgery, as well as all the Medicines, the Patient is to be charged for, after it has been through the *InputErrorControl* function.
    - Type: char
  + *Days*
    - A variable to store the input of the User, as to how many days the Patient spent in Hospital care.
    - Type: int
  + *Choice\_2*
    - A variable to store the selection of the User, as to whether they wish to enter the Charges for another Patient or not.
    - Type: string
  + *Account\_Setup*
    - Object to access the class *patientAccountType.*
  + *Input*
    - A variable in the function *InputErrorControl* to store the selection of the User, with regards to Medicine and Surgery Type, and correct it so that it is interpreted correctly by the program.
    - Type: char

**Program Design:**

1. Prompt the User to select what kind of Surgery the Patient has undergone.
2. Prompt the User to select what kind of Medicines the Patient has received. Ask them to enter All those administered, but one at a time.
3. Prompt the User to insert the Number of Days the Patient has spent at the Hospital.
4. Display the Total Cost, along with the Rate for a Single Day spent at the Hospital and the Number of Days that were received by the Program. Moreover, also display the Total Cost for the Surgery and Medicines.
5. Prompt the User to enter “Yes” or “No”, as to if they wish to enter the Costs for another Patient. If “Yes”, Repeat the Process. If “No”, terminate the program.

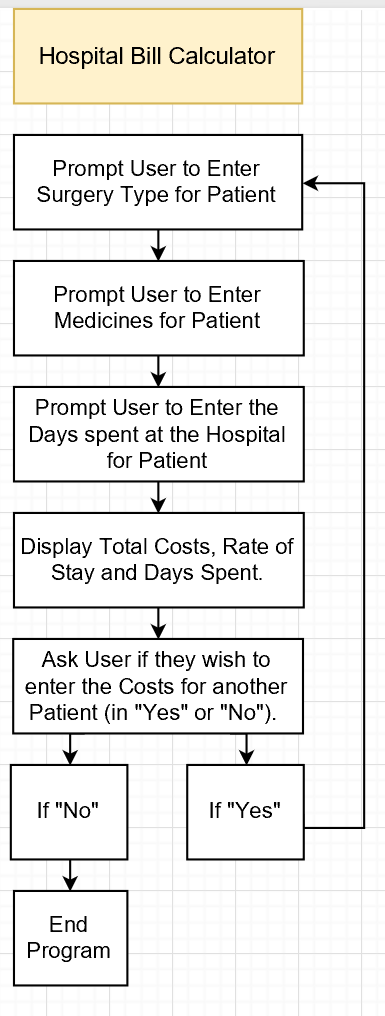
 **Flowchart:**

Figure. A

**Algorithm Development (Detailed Pseudocode):**

* Set *Account\_Setup* as the object to access the class *patientAccountType..*
* Setup a do-While loop that includes the following:
  + Prompt the User to insert the Type of Surgery that the Patient has undergone, through a Menu, that looks like the following:
    - Hospital Bill Menu
    - ---------------------------------------
    - Surgery Selection
    - ---------------------------------------
    - A - Arm Surgery
    - B - Leg Surgery
    - C - Eye Surgery
    - D - Brain Surgery
    - E - Heart Surgery
    - Select the Type of Surgery, undertaken by Patient (A, B, C, D or E):-

Store the Users input in the variable *Choice*.

* + Run the information in *Choice*, through the function *InputErrorControl(Choice)*, which will convert lowercase into uppercase (‘a’ into ‘A’, ‘b’ into ‘B’), so that the Program can interpret the choice of the User correctly. Store this in the variable *Final\_Choice*.
  + Run the information in *Final\_Choice*, through the function *Account\_Setup.setSurgeryCharges(Final\_Choice)*. This will add the cost of the surgery to the private variable *totalCharges* in the class *patientAccountType*.
  + Setup another do-While loop that includes the following:
    - Prompt the User to insert the Medication that the Patient has received, through a Menu, that looks like the following:
      * ---------------------------------------
      * Medicine Selection
      * ---------------------------------------
      * A - Med\_A
      * B - Med\_B
      * C - Med\_C
      * D - Med\_D
      * E - Med\_E
      * F - Exit Medicine Menu
      * Enter ALL Medications Received/Prescribed to the Patient (Enter 1 Medicine at a Time, then Enter 'F' when done):-

Store the Users input in the variable *Choice*.

* + - Run the information in *Choice*, through the function *InputErrorControl(Choice)*, which will convert lowercase into uppercase (‘a’ into ‘A’, ‘b’ into ‘B’), so that the Program can interpret the choice of the User correctly. Store this in the variable *Final\_Choice*.
    - Run the information in *Final\_Choice*, through the function *Account\_Setup.setPharmacyCharges(Final\_Choice)*. This will add the cost of the surgery to the private variable *totalCharges* in the class *patientAccountType*.
  + End the do-While loop when *Final\_Choice* equal to ‘F’. (In other words, Run do-While loop for as long as *Final\_Choice* not equal to ‘F’.)
  + Prompt the User to insert the Number of Days that the Patient has spent at the Hospital, through a Menu, that looks like the following:
    - ---------------------------------------
    - Time of Stay
    - ---------------------------------------
    - Enter the Number of Days the Patient was admitted for at the Hospital:-

Store the Users input in the variable *Days*.

* + Run the information in *Days*, through the function *Account\_Setup.setDays(Days)*. This will set the private variable *Days*, in the class *patientAccountType*, to the value inserted by the User and allow for the computation of the Total Overall Cost for the Patient later.
  + Display the Results through the following:
    - ---------------------------------------
    - Total Report
    - ---------------------------------------
    - Number of Days spent, in the Hospital:-

Retrieve this Information through the function *Account\_Setup.getDays()*. This will get the Number of Days entered by the User, which are stored in the private variable *Days*.

* + - Hospital Occupancy Charge:-

Retrieve this Information through the function *Account\_Setup.getRate()*. This will get the Amount that is charged for each day spent at the Hospital, which is stored in the private variable *Rate*.

* + - Total Charge for Surgery and Medicine:-

Retrieve this Information through the function *Account\_Setup.getTotalCharges()*. This will get the combined value of the cost for the Surgery and the Medications, stored in the private variable *totalCharges*.

* + - Total Cost at Checkout:

Retrieve this Information through the function *Account\_Setup.getCost()*. This will get the combined value of the cost for the Surgery and the Medications, as well as the Hospital stay charges through the computation, ( (*Days* \* *Rate*) + *totalCharges*).

* + Reset the Value of *totalCharges* to 0, through the function *Account\_Setup.setTotalCharges(0)*.
  + Prompt the User to insert “Yes” or “No”, for whether or not they wish to input the Costs for another Patient, Through the following prompt:
    - Do you want to Compute the Costs, for another Patient (Enter Yes or No):-

Store the Users input in the variable *Choice\_2*.

* Based on *Choice\_2*, end the do-While loop or not. If *Choice\_2* is “Yes”, repeat the whole process. If *Choice\_2* is “No”, end the program.

**Program Listing:-**

***surgeryType.h:***

1. #ifndef SURGERYTYPE\_H
2. #define SURGERYTYPE\_H
3. class surgeryType
4. {
5. public:
6. /\*Function to retrieve the information stored in the private variable Arm\*/
7. int getArm();
8. /\*Function to retrieve the information stored in the private variable Leg\*/
9. int getLeg();
10. /\*Function to retrieve the information stored in the private variable Eye\*/
11. int getEye();
12. /\*Function to retrieve the information stored in the private variable Brain\*/
13. int getBrain();
14. /\*Function to retrieve the information stored in the private variable Heart\*/
15. int getHeart();
16. /\*Constructor\*/
17. surgeryType();
18. private:
19. /\*Variable to store the Cost of Arm Surgery\*/
20. int Arm;
21. /\*Variable to store the Cost of Leg Surgery\*/
22. int Leg;
23. /\*Variable to store the Cost of Eye Surgery\*/
24. int Eye;
25. /\*Variable to store the Cost of Brain Surgery\*/
26. int Brain;
27. /\*Variable to store the Cost of Heart Surgery\*/
28. int Heart;
29. };
30. #endif // SURGERYTYPE\_H

***surgeryType.cpp:***

1. #include "surgeryType.h"
2. int surgeryType::getArm()
3. {
4. return Arm;
5. }
6. int surgeryType::getLeg()
7. {
8. return Leg;
9. }
10. int surgeryType::getEye()
11. {
12. return Eye;
13. }
14. int surgeryType::getBrain()
15. {
16. return Brain;
17. }
18. int surgeryType::getHeart()
19. {
20. return Heart;
21. }
22. surgeryType::surgeryType()
23. {
24. Arm = 13040;
25. Leg = 18110;
26. Eye = 19705;
27. Brain = 39750;
28. Heart = 28965;
29. }

***pharmacyType.h:***

1. #ifndef PHARMACYTYPE\_H
2. #define PHARMACYTYPE\_H
3. class pharmacyType
4. {
5. public:
6. /\*Function to retrieve the information stored in the private variable Med\_A\*/
7. int getMed\_A();
8. /\*Function to retrieve the information stored in the private variable Med\_B\*/
9. int getMed\_B();
10. /\*Function to retrieve the information stored in the private variable Med\_C\*/
11. int getMed\_C();
12. /\*Function to retrieve the information stored in the private variable Med\_D\*/
13. int getMed\_D();
14. /\*Function to retrieve the information stored in the private variable Med\_E\*/
15. int getMed\_E();
16. /\*Constructor\*/
17. pharmacyType();
18. private:
19. /\*Variable to store the Cost of Med\_A\*/
20. int Med\_A;
21. /\*Variable to store the Cost of Med\_B\*/
22. int Med\_B;
23. /\*Variable to store the Cost of Med\_C\*/
24. int Med\_C;
25. /\*Variable to store the Cost of Med\_D\*/
26. int Med\_D;
27. /\*Variable to store the Cost of Med\_E\*/
28. int Med\_E;
29. };
30. #endif // PHARMACYTYPE\_H

***pharmacyType.cpp:***

1. #include "pharmacyType.h"
2. int pharmacyType::getMed\_A()
3. {
4. return Med\_A;
5. }
6. int pharmacyType::getMed\_B()
7. {
8. return Med\_B;
9. }
10. int pharmacyType::getMed\_C()
11. {
12. return Med\_C;
13. }
14. int pharmacyType::getMed\_D()
15. {
16. return Med\_D;
17. }
18. int pharmacyType::getMed\_E()
19. {
20. return Med\_E;
21. }
22. pharmacyType::pharmacyType()
23. {
24. Med\_A = 811;
25. Med\_B = 108;
26. Med\_C = 901;
27. Med\_D = 808;
28. Med\_E = 734;
29. }

***patientAccountType.h:***

1. #ifndef PATIENTACCOUNTTYPE\_H
2. #define PATIENTACCOUNTTYPE\_H
3. #include "surgeryType.h"
4. #include "pharmacyType.h"
5. class patientAccountType
6. {
7. public:
8. /\*Function to add the cost of the Medicines to the private variable totalCharges\*/
9. void setPharmacyCharges(char Option);
10. /\*Function to add the cost of the Surgery to the private variable totalCharges\*/
11. void setSurgeryCharges (char Option);
12. /\*Function to retrieve the total cost of the Surgery, Medicines and Hospital stay combined\*/
13. int getCost();
14. /\*Function to retrieve the information stored in the private variable Rate\*/
15. int getRate();
16. /\*Function to retrieve the information stored in the private variable totalCharges\*/
17. int getTotalCharges();
18. /\*Function to retrieve the information stored in the private variable Days\*/
19. int getDays();
20. /\*Function to set the value of private variable Rate\*/
21. void setRate (int R);
22. /\*Function to set the value of private variable totalCharges\*/
23. void setTotalCharges(int Total);
24. /\*Function to set the value of private variable Days \*/
25. void setDays (int D);
26. /\*Constructor with Parameters\*/
27. patientAccountType(int D, int Total, int R);
28. /\*Constructor\*/
29. patientAccountType();
30. private:
31. /\*Variable to store the Days spent by the Patient at the Hospital\*/
32. int Days;
33. /\*Variable to store the charges for the Medication and Surgery\*/
34. int totalCharges;
35. /\*Variable to store the Amount that is charged for each day spent at the Hospital\*/
36. int Rate;
37. /\*Object to access the class surgeryType\*/
38. surgeryType surgeryCharges;
39. /\*Object to access the class pharmacyType\*/
40. pharmacyType pharmacyCharges;
41. };
42. #endif // PATIENTACCOUNTTYPE\_H

***patientAccountType.cpp:***

1. #include "patientAccountType.h"
2. patientAccountType::patientAccountType()
3. {
4. Days = 1;
5. Rate = 100;
6. totalCharges = 0;
7. }
8. patientAccountType::patientAccountType(int D, int Total, int R)
9. {
10. Days = D;
11. Rate = R;
12. totalCharges = Total;
13. }
14. void patientAccountType::setDays(int D)
15. {
16. Days = D;
17. }
18. void patientAccountType::setTotalCharges(int Total)
19. {
20. totalCharges = Total;
21. }
22. void patientAccountType::setRate(int R)
23. {
24. Rate = R;
25. }
26. int patientAccountType::getDays()
27. {
28. return Days;
29. }
30. int patientAccountType::getTotalCharges()
31. {
32. return totalCharges;
33. }
34. int patientAccountType::getRate()
35. {
36. return Rate;
37. }
38. int patientAccountType::getCost()
39. {
40. int A;
41. A = ((Days\*Rate)+totalCharges);
42. return A;
43. }
44. void patientAccountType::setSurgeryCharges(char Option)
45. {
46. if(Option == 'A')
47. {
48. totalCharges += surgeryCharges.getArm();
49. }
50. else if(Option == 'B')
51. {
52. totalCharges += surgeryCharges.getLeg();
53. }
54. else if(Option == 'C')
55. {
56. totalCharges += surgeryCharges.getEye();
57. }
58. else if(Option == 'D')
59. {
60. totalCharges += surgeryCharges.getBrain();
61. }
62. else if(Option == 'E')
63. {
64. totalCharges += surgeryCharges.getHeart();
65. }
66. }
67. void patientAccountType::setPharmacyCharges(char Option)
68. {
69. if(Option == 'A')
70. {
71. totalCharges += pharmacyCharges.getMed\_A();
72. }
73. else if(Option == 'B')
74. {
75. totalCharges += pharmacyCharges.getMed\_B();
76. }
77. else if(Option == 'C')
78. {
79. totalCharges += pharmacyCharges.getMed\_C();
80. }
81. else if(Option == 'D')
82. {
83. totalCharges += pharmacyCharges.getMed\_D();
84. }
85. else if(Option == 'E')
86. {
87. totalCharges += pharmacyCharges.getMed\_E();
88. }
89. }

***Main Program:***

1. /\*(Three Classes)Hospital Bill Calculator by Hissamuddin Shaikh\*/
2. #include <iostream>
3. #include <iomanip>
4. #include <string>
5. #include "surgeryType.h"
6. #include "pharmacyType.h"
7. #include "patientAccountType.h"
8. using namespace std;
9. /\*Prototype for Error Control Function\*/
10. char InputErrorControl(char);
11. /\*Main Function\*/
12. int main(){
13. /\*Variable Setup\*/
14. char Choice;
15. char Final\_Choice;
16. int Days;
17. string Choice\_2;
18. patientAccountType Account\_Setup;
19. /\*Setup for do-While loop of Patient Costs\*/
20. do{
21. /\*Set up for Surgery Type Selection and Cost Calculation\*/
22. cout<< endl << setw(29) << "Hospital Bill Menu" << endl;
23. cout<< endl << setw(40) << "---------------------------------------";
24. cout<< endl << setw(28) << "Surgery Selection";
25. cout<< endl << setw(40) << "---------------------------------------" << endl;
26. cout<< endl << setw(17) << "A - Arm Surgery"<< endl;
27. cout<< endl << setw(17) << "B - Leg Surgery"<< endl;
28. cout<< endl << setw(17) << "C - Eye surgery"<< endl;
29. cout<< endl << setw(19) << "D - Brain Surgery"<< endl;
30. cout<< endl << setw(19) << "E - Heart Surgery"<< endl;
31. cout<< endl << setw(72) << "Select the Type of Surgery, undertaken by Patient (A, B, C, D or E):- ";
32. cin>> Choice;
33. Final\_Choice = InputErrorControl(Choice);
34. Account\_Setup.setSurgeryCharges(Final\_Choice);
35. /\*Setup for a do-While loop to take in Multiple Medicines\*/
36. do{
37. /\*Set up for Medicine Selection and Cost Calculation\*/
38. cout<< endl << setw(40) << "---------------------------------------";
39. cout<< endl << setw(28) << "Medicine Selection";
40. cout<< endl << setw(40) << "---------------------------------------" << endl;
41. cout<< endl << setw(11) << "A - Med\_A" <<endl;
42. cout<< endl << setw(11) << "B - Med\_B" <<endl;
43. cout<< endl << setw(11) << "C - Med\_C" <<endl;
44. cout<< endl << setw(11) << "D - Med\_D" <<endl;
45. cout<< endl << setw(11) << "E - Med\_E" <<endl;
46. cout<< endl << setw(24) << "F - Exit Medicine Menu" <<endl;
47. cout<< endl << setw(59) << "Enter ALL Medications Received/Prescribed to the Patient "
48. << "(Enter 1 Medicine at a Time, then Enter 'F' when done):- ";
49. cin>>Choice;
50. Final\_Choice = InputErrorControl(Choice);
51. Account\_Setup.setPharmacyCharges(Final\_Choice);
52. /\*End Condition for nested do-While loop\*/
53. }while(Final\_Choice != 'F');
54. /\*Set up for Prompting Days of Stay at Hospital\*/
55. cout<< endl << setw(40) << "---------------------------------------";
56. cout<< endl << setw(25) << "Time of Stay";
57. cout<< endl << setw(40) << "---------------------------------------" << endl;
58. cout<< endl << setw(74) << "Enter the Number of Days the Patient was admitted for at the Hospital:- ";
59. cin>>Days;
60. Account\_Setup.setDays(Days);
61. /\*Setup for Displaying Results\*/
62. cout<< endl << setw(40) << "---------------------------------------";
63. cout<< endl << setw(25) << "Total Report";
64. cout<< endl << setw(40) << "---------------------------------------" << endl;
65. cout<< endl << setw(42) << "Number of Days spent, in the Hospital:- "<< Account\_Setup.getDays()<<endl;
66. cout<< endl << setw(30) << "Hospital Occupancy Charge:- "<< Account\_Setup.getRate()<<endl;
67. cout<< endl << setw(42) << "Total Charge for Surgery and Medicine:- "<< Account\_Setup.getTotalCharges()<<endl;
68. cout<< endl << setw(26) << "Total Cost at Checkout: "<< Account\_Setup.getCost()<<endl;
69. /\*Setup for reseting Total Charges for the potential Patient\*/
70. Account\_Setup.setTotalCharges(0);
71. /\*Set up for prompting user to enter New Patient or Not, also the End Condition for main do-While loop\*/
72. cout<< endl << setw(76) << "Do you want to Compute the Costs, for another Patient (Enter Yes or No):- ";
73. cin>> Choice\_2;
74. cout<< endl << setw(40) << "---------------------------------------" << endl;
75. /\*Error Control for Choice\_2\*/
76. if (Choice\_2=="Yes"||Choice\_2=="yes"||Choice\_2=="YES"||Choice\_2=="YEs"||Choice\_2=="yES"||Choice\_2=="yEs"||Choice\_2=="yeS")
77. Choice\_2 = "Yes";
78. } while(Choice\_2 == "Yes");
79. return 0;
80. }
81. /\*Setup for Error Control Function\*/
82. char InputErrorControl(char Input)
83. {
84. if (Input == 'a')
85. Input = 'A';
86. else if (Input == 'b')
87. Input = 'B';
88. else if (Input == 'c')
89. Input = 'C';
90. else if (Input == 'd')
91. Input = 'D';
92. else if (Input == 'e')
93. Input = 'E';
94. else if (Input == 'f')
95. Input = 'F';
96. return Input;
97. }

**Note:**

Screenshots of the original code are attached at the end of the document (in the notes section), incase this is unclear, or something didn’t copy appropriately.

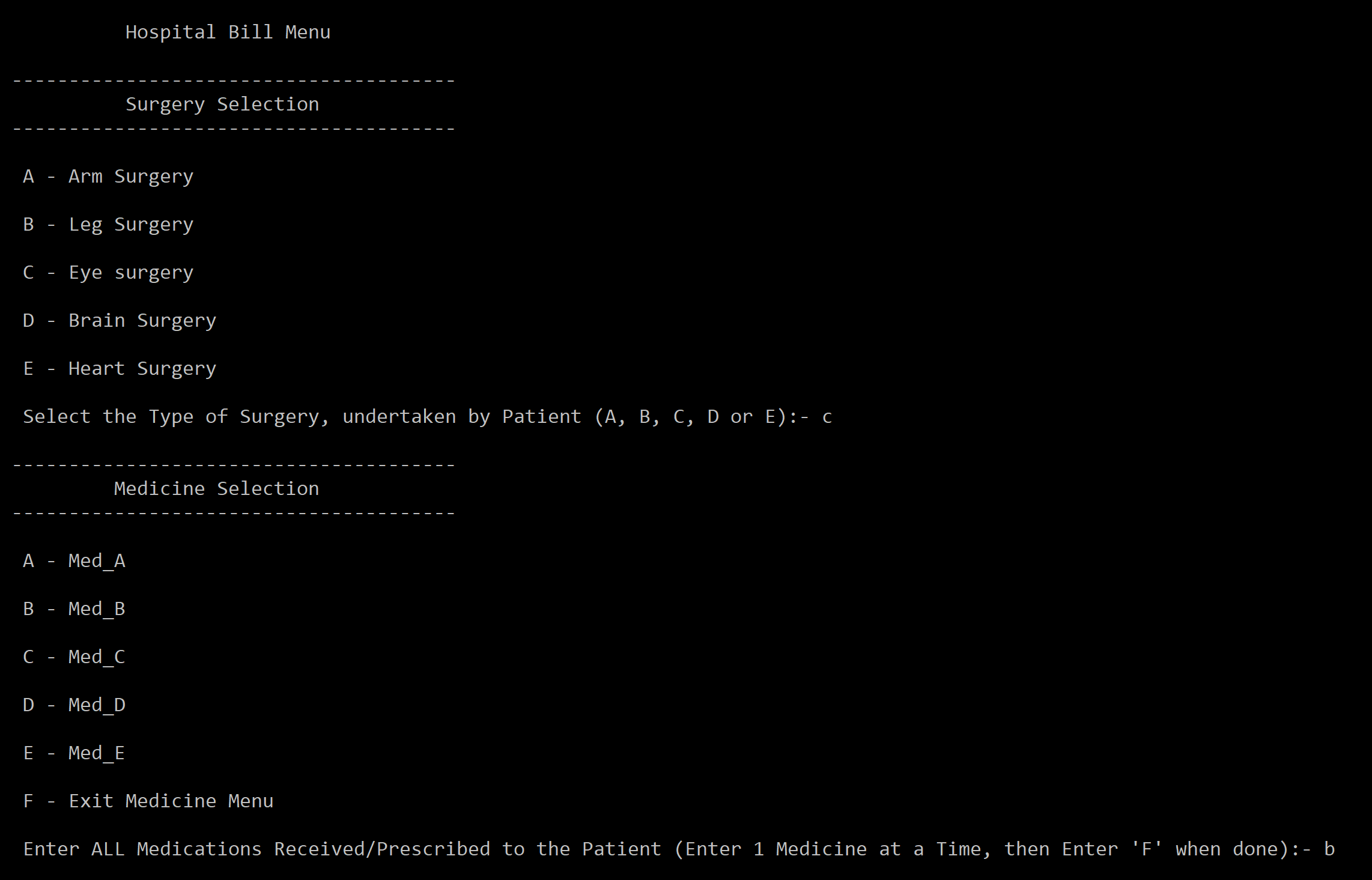
**Sample test run of the program:**

A screenshot of a cell phone

Description generated with very high confidence 1st Sample:

2nd Sample:

A screenshot of a cell phone

Description generated with very high confidence

A screenshot of a cell phone

Description generated with very high confidence

**Observations, error handling and general comments:**

The program runs correctly as the result in the 1st Sample and 2nd Sample, correspond to manual computations.

**1st Sample (Manual Computation) =**

Selected Surgery = a (Arm Surgery)

Selected Medicine = No Medicine (f)

Number of Days = 2

Cost of Surgery = 13040

Cost of Medicine = 0

Cost of Hospital Occupancy = 200

Total Charge for Surgery and Medicine = 13040

Total Cost at Checkout = 13240

Repetition (for other Patients) = No

This is the same as the one computed by the computer.

**2nd Sample (Manual Computation) =**

Selected Surgery = c (Eye Surgery)

Selected Medicine = b, c (Med\_B, Med\_C)

Number of Days = 5

Cost of Surgery = 19705

Cost of Medicine = 108, 901

Cost of Hospital Occupancy = 500

Total Charge for Surgery and Medicine = 20714

Total Cost at Checkout = 21214

Repetition (for other Patients) = no

This is the same as the one computed by the computer.

No errors should be encountered, if the user follows the prompts, and enters data in the appropriate format (no decimals, and numbers only where numbers are required). Moreover, various measures have been taken to limit the input of error causing data, such as converting the user input to the correct format through if-statements (‘a’ to ‘A’, ‘b’ to ‘B’….). Additionally, if the user enters “Yes” in any form (yes, YES, Yes…), it will also be converted to the proper format through an if-statement. Therefore, a lot of error control was employed in this program, all of which cannot be shown as the report would become excessively long. Not to mention, the presentation of the Program was made as neat and comprehensible as possible.

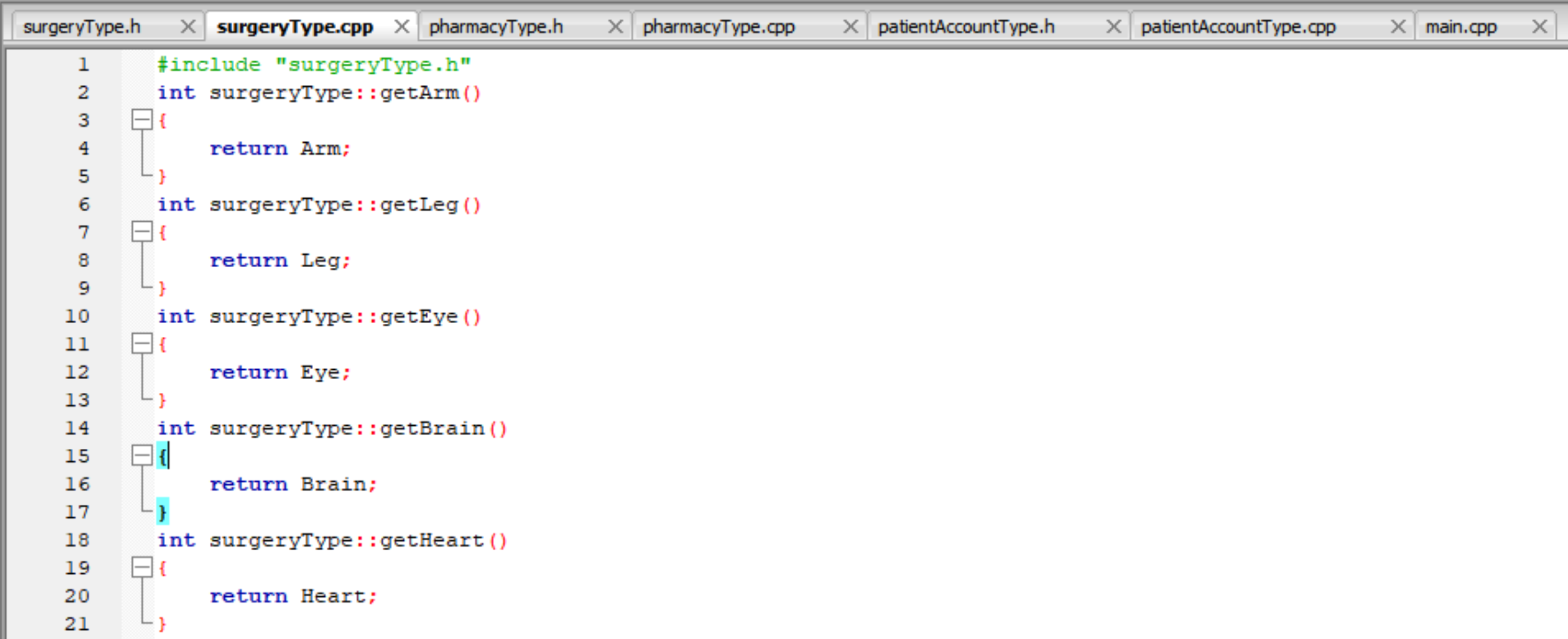
**Conclusions**

The Hospital Bill Calculator makes great use of the three classes, surgeryType, pharmacyType and patientAccountType, as it constructs a very realistic scenario of a Hospital Bill calculation and construction. I do acknowledge that some of the member functions in these classes were not utilized, but they do function correctly. It’s just that the implementation of all member functions in a single program, is far too difficult and unlikely. Therefore, the Hospital Bill Calculator is a well-rounded program, capable of performing its task effectively.

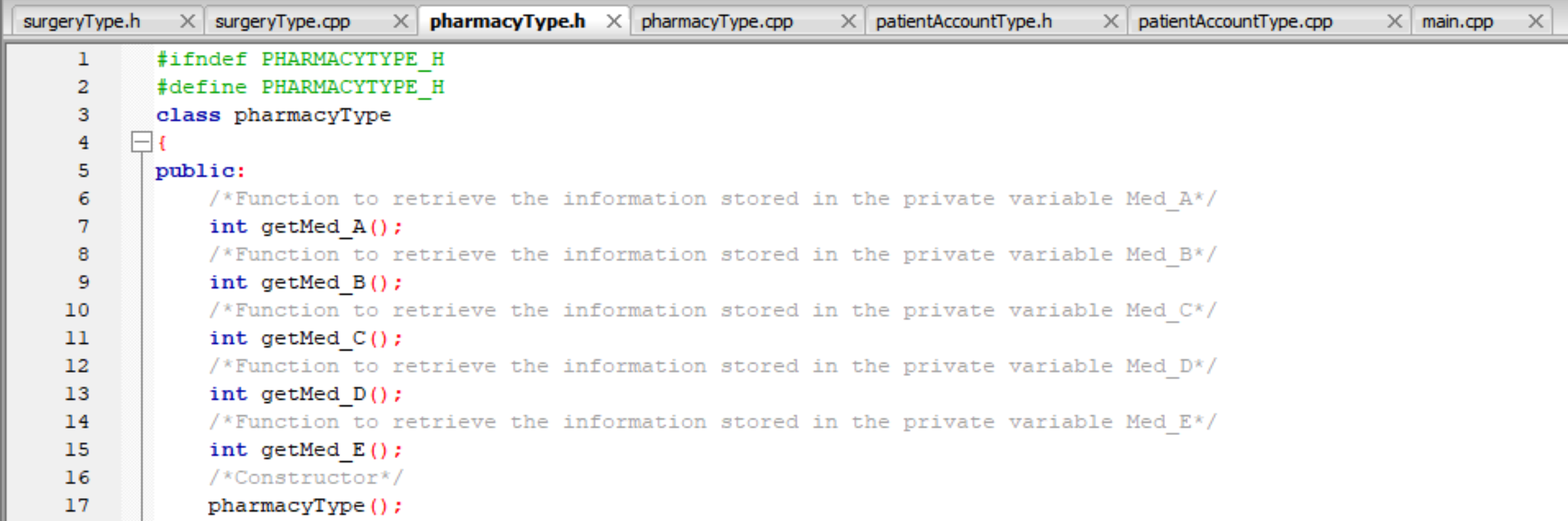
**Notes:**

A screenshot of a cell phone

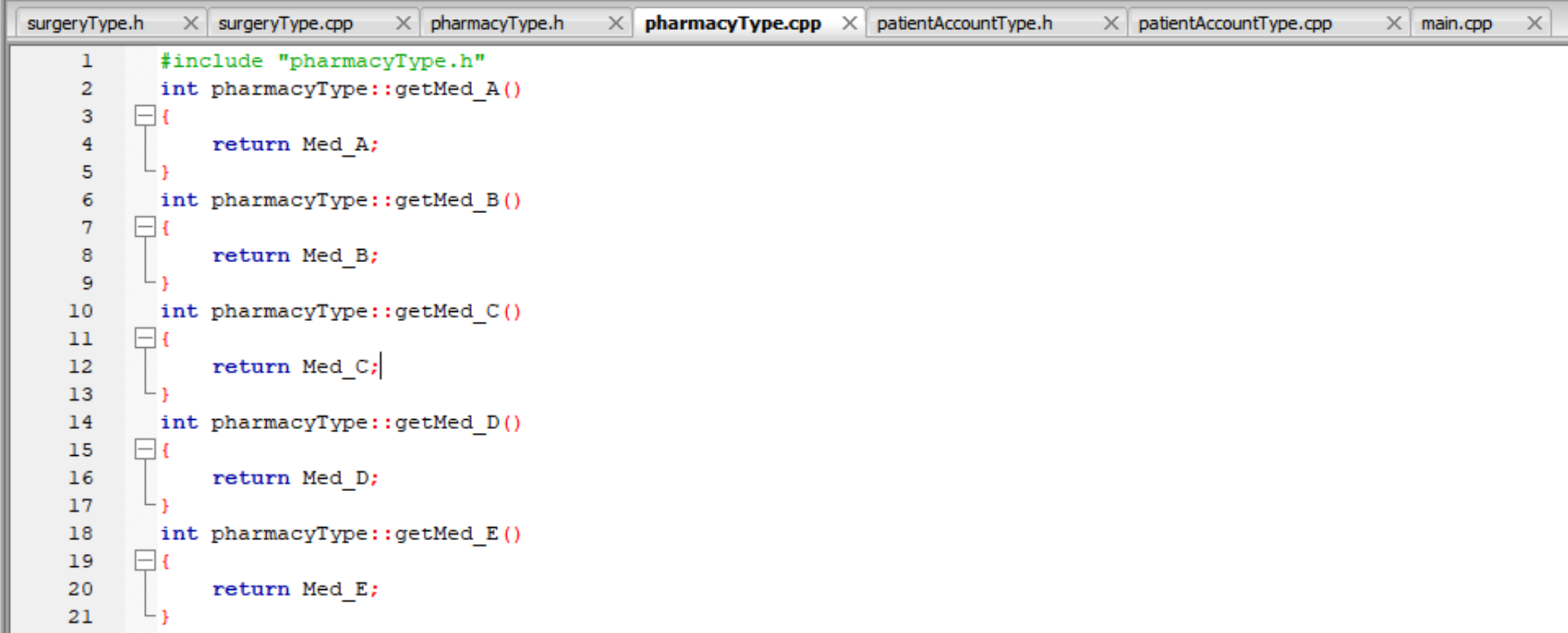
Description generated with very high confidence

A screenshot of a cell phone

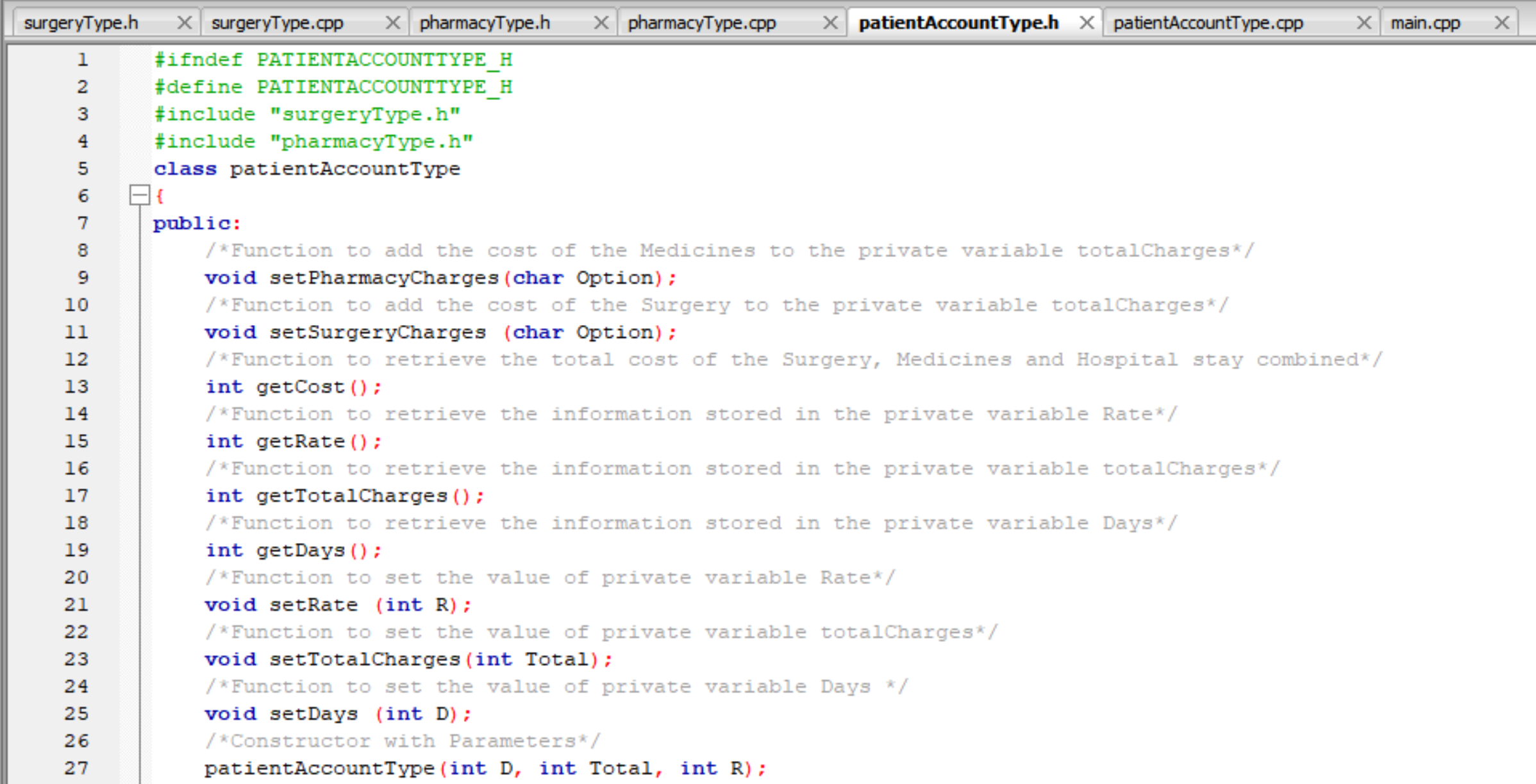
Description generated with very high confidence

A screenshot of a cell phone

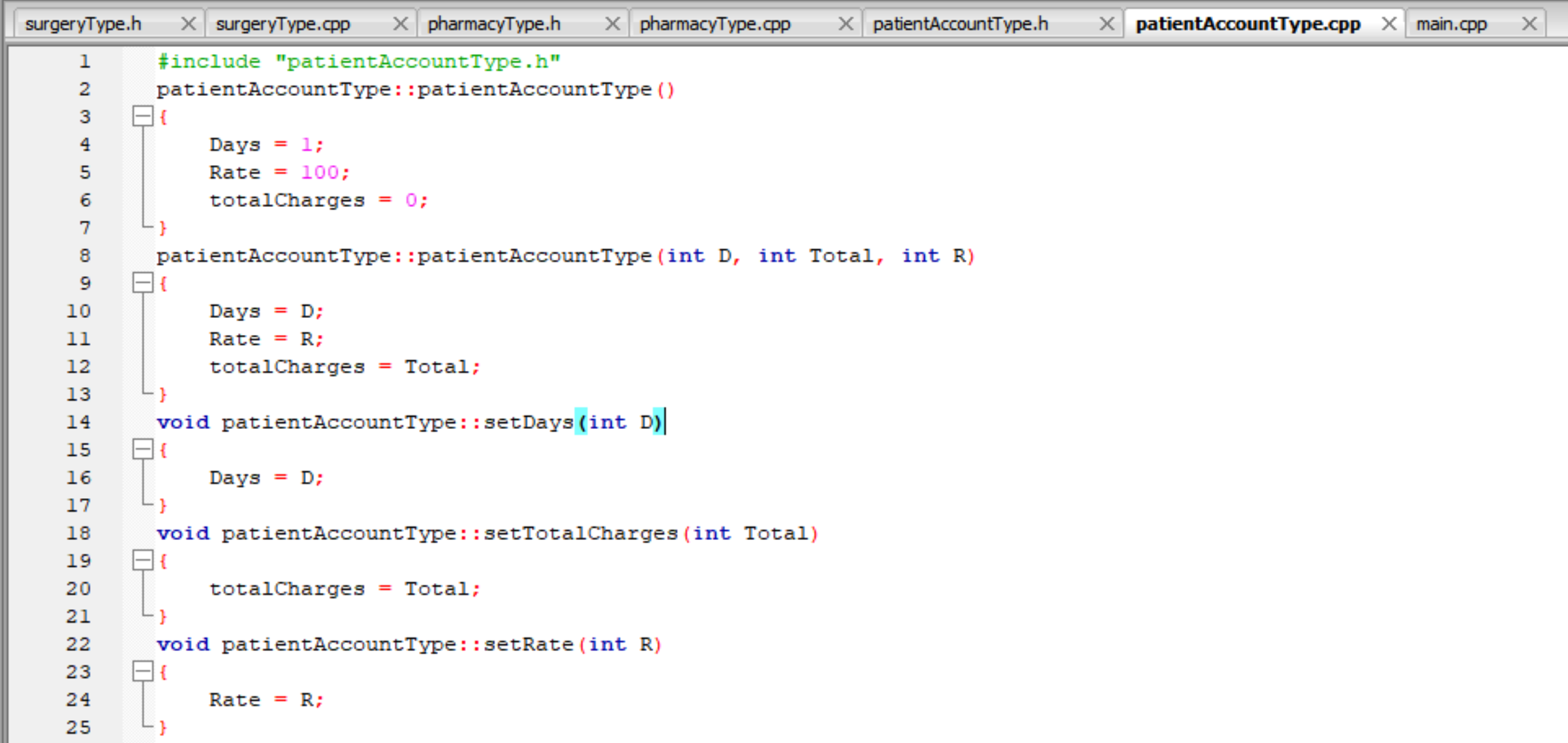
Description generated with very high confidence

A screenshot of a cell phone

Description generated with very high confidence

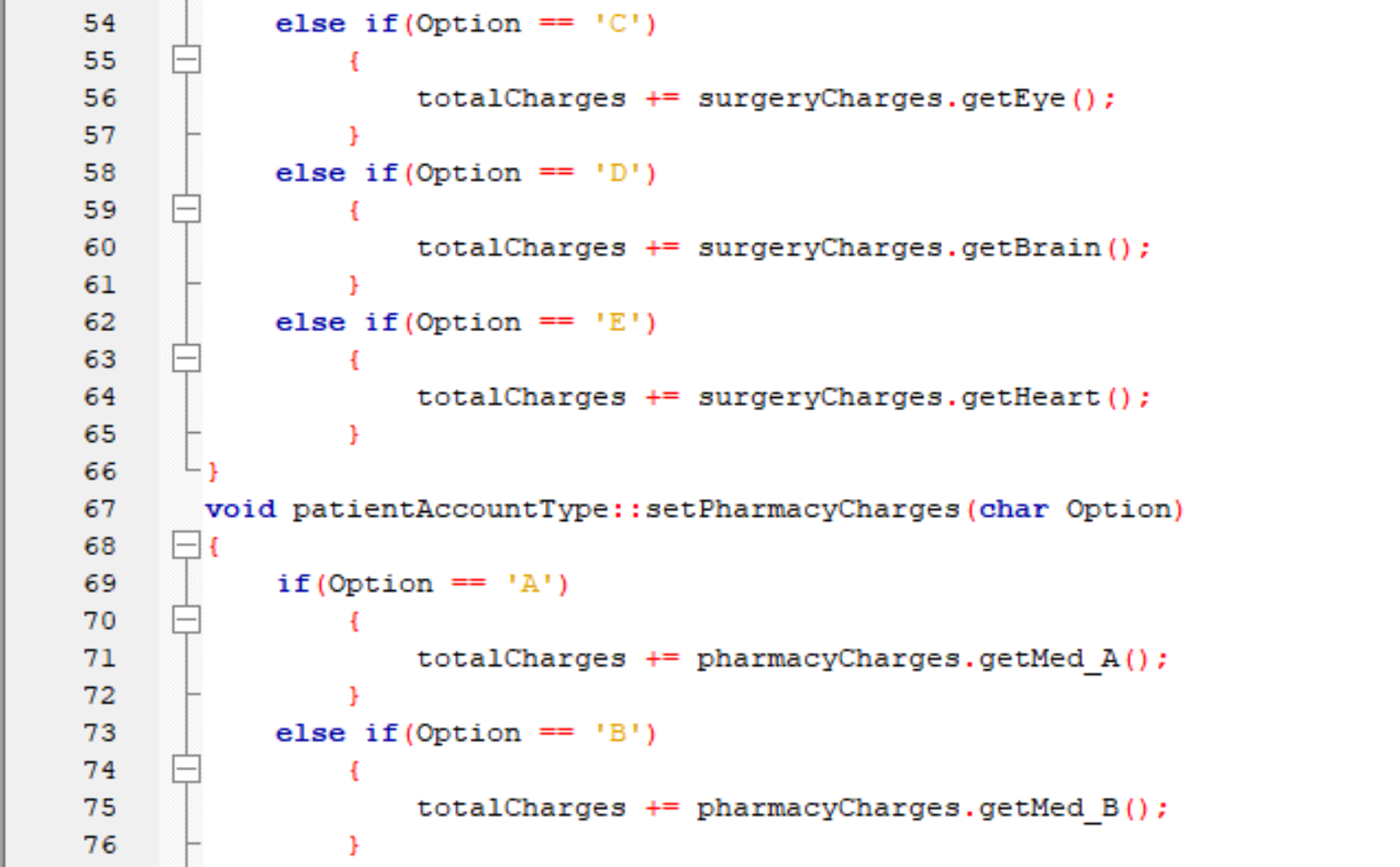
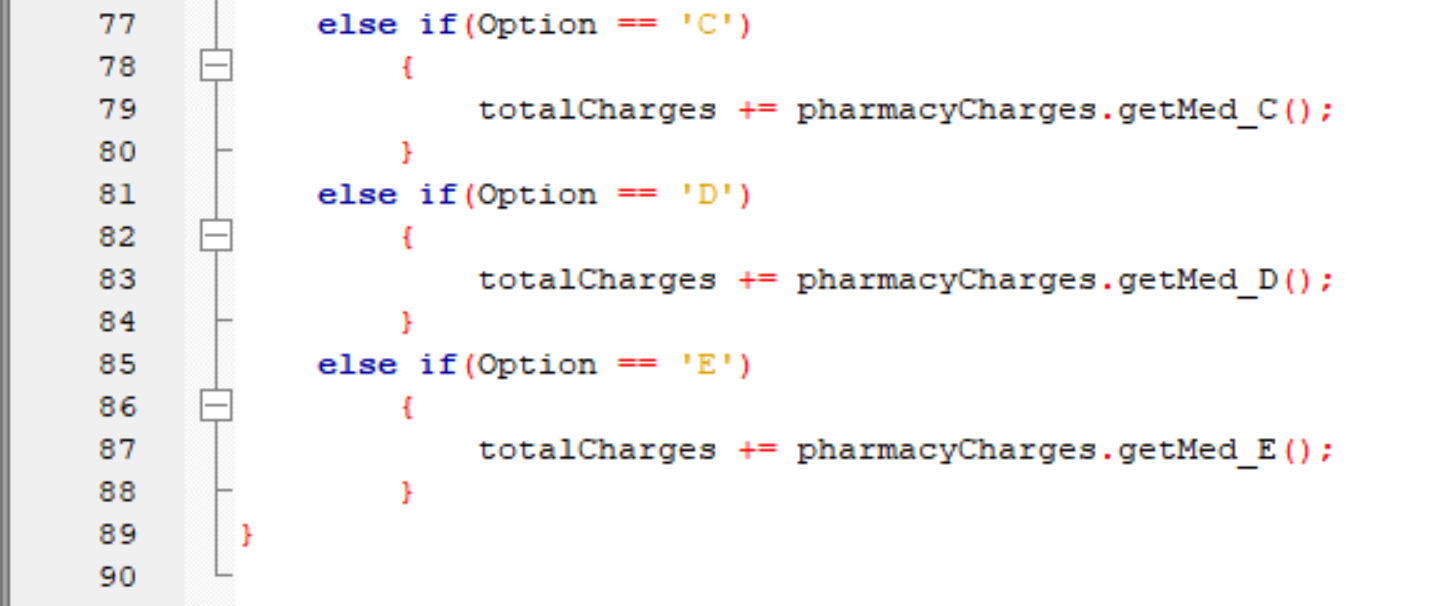
A screenshot of a social media post

Description generated with very high confidence



A screenshot of a social media post

Description generated with very high confidence



A screenshot of a social media post

Description generated with very high confidence

A screenshot of a social media post

Description generated with very high confidence

A screenshot of a social media post

Description generated with very high confidence

A screenshot of a cell phone

Description generated with very high confidence