

coding

Dentistry

4 MIGRANT WOMEN RETURNERS

Booking System

Advanced



Activity scenario summary:

In this example we will code a project, which replicates a simple **Dentistry Booking System**, used in dental practices.

This project will help you develop an understanding how coding is used in dentistry. As part of the activity participants will design a software in which the dentistry staff will be able to;

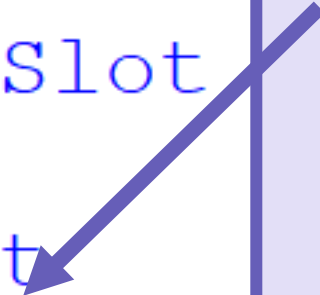
- a) Book Appointments
- b) Clear Appointment Slots
- c) View Booking Lists
- d) View Procedure Lists
- e) Calculate Bill

In this part of the project, we are going to allow the user to calculate a bill for their patient

Find the finished code for this project [here](#) - use this if you get stuck!

Calculate Bill

```
-- Menu --  
1) Book Appointment  
2) Clear Appointment Slot  
3) View Booking List  
4) View Procedure List  
5) Calculate Bill  
Enter an option: |
```



To calculate bill:

- Ask the user to enter procedures that have been performed
- Add money to the bill for each procedure
- Add appointment charge
- Display final bill

Calculate Bill

```
--- Procedures ---
1) Cleaning      50
2) Whitening     70
3) Extractions  120
4) Veneers       310
5) Filling        40
6) Crowns        180
7) Root Canal    440
8) Braces        870
```

```
--- Calculating Bill ---
Enter ! to finish
Select a procedure: 1
Select a procedure: 6
Select a procedure: 7
Select a procedure: !

--- Bill ---
- Appointment Charge  50
- Cleaning            50
- Crowns              180
- Root Canal          440

Total:  720
```

To calculate bill:

- Ask the user to enter procedures that have been performed
- Add money to the bill for each procedure
- Add appointment charge
- Display final bill

Step 1

Creating a subroutine...

```
62 #Calculate bill
63 def calculateBill():
64     total = 0
65     notCompleted = True
66     viewProceduresList()
67
68     print("\n --- Calculating Bill ---")
69     print("Enter ! to finish")
70
```

Define a subroutine and declare 3 variables.

Also, call the viewProceduresList() subroutine so the user can see what to choose from!

Print out some labels so the user can see what's happening!

Step 2

Getting procedures from the user...

Create a WHILE LOOP - this will repeatedly ask the user to enter procedures until the user enters !

```
70
71 while notCompleted:
72     option = input("Select a procedure: ")
73
74     if option == "1":
75         total += procedureCost[0]
76         bill.append([procedures[0], procedureCost[0]])
77
```

We then create an IF statement - if the user enters 1, the program will add the cost of the first procedure to our total and add the name of the procedure to the bill list

Step 3

Expanding our IF Statement..

We want to repeat this process by adding ELIF statements - there should be one of these for every procedure option!

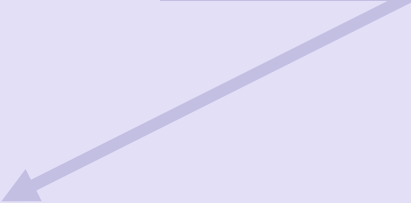
```
78 elif option == "2":  
79     total += procedureCost[1]  
80     bill.append([procedures[1], procedureCost[1]])  
81  
82 elif option == "3":  
83     total += procedureCost[2]  
84     bill.append([procedures[2], procedureCost[2]])  
85
```

Step 3

Expanding our IF Statement..

```
if option == "1":  
    total += procedureCost[0]  
    bill.append([procedures[0],procedureCost[0]])  
  
elif option == "2":  
    total += procedureCost[1]  
    bill.append([procedures[1],procedureCost[1]])  
  
elif option == "3":  
    total += procedureCost[2]  
    bill.append([procedures[2],procedureCost[2]])  
  
elif option == "4":  
    total += procedureCost[3]  
    bill.append([procedures[3],procedureCost[3]])  
  
elif option == "5":  
    total += procedureCost[4]  
    bill.append([procedures[4],procedureCost[4]])  
  
elif option == "6":  
    total += procedureCost[5]  
    bill.append([procedures[5],procedureCost[5]])  
  
elif option == "7":  
    total += procedureCost[6]  
    bill.append([procedures[6],procedureCost[6]])  
  
elif option == "8":  
    total += procedureCost[7]  
    bill.append([procedures[7],procedureCost[7]])
```

Eventually it should look like this!



Step 4

Allowing the user to stop..

```
elif option == "8":  
    total += procedureCost[7]  
    bill.append([procedures[7], procedureCost[7]])  
  
elif option == "!":  
    notCompleted = False  
else:  
    print("Option not available. Try again.")
```

We need to add a few more clauses to our IF ELIF statement.

This ELIF will allow the user to finish entering procedures - by entering a '!' the loop will stop.

This ELSE statement will run if the user enters a random number that isn't on the procedure list

Step 5

Finishing our bill and printing it out...

```
total += appointmentCharge

print("\n --- Bill --- ")
print(" - Appointment Charge\t" + str(appointmentCharge))
for i in range(len(bill)):
    print(" - " + bill[i][0] + "\t" + str(bill[i][1]))
print("\nTotal:\t" + str(total))
```

To finish off our subroutine, add on the appointmentCharge to the total

Again, try and print out labels so your user knows what's happening!

Then, using a FOR loop, we go through every procedure the user has selected, printing out the name and price.

Finally, display the total!

Step 6

Update our menu...

```
while True:
    print("\n-- Menu --")
    print("1) Book Appointment")
    print("2) Clear Appointment Slot")
    print("3) View Booking List")
    print("4) View Procedure List")
    print("5) Calculate Bill")

    option = input("Enter an option: ")

    if option == "1":
        bookAppointment()
    elif option == "2":
        clearAppointmentSlot()
    elif option == "3":
        viewBookingList()
    elif option == "4":
        viewProceduresList()
    elif option == "5":
        calculateBill()
    else:
        print("Input error - try again.")
```

Finally, update the menu by calling the `calculateBill()` whenever the number 5 is selected

Congratulations!

**You have completed the dentistry
booking software.**

4 MIGRANT WOMEN RETURNERS

Find the finished code for this project [here](#) - use this if you get stuck!