|  |
| --- |
| Student #: 1283 701 40 |
| CP1401 – Part 2 |
| Tropic Airlines Ticket Ordering System |

|  |
| --- |
| Cameron Clark  5-14-2016 |

**Functions:**

greeting():  
  *appears on startup, asks for users name. Response saved as userName.*

getAge():  
 *gets age from the user, ensure is within expected range*

checkHandle(prompt,codeList,wordList):  
 a relevant prompt, codeList and wordlist is passed into the function

returns the relevant word form code and workList

costCalculation(tripDestination, flightType, seatType, seatClass, passengerAge):

*Calculates cost of trip from expected values*

acceptPurchase(cost):  
  *presents user with a y/n to purchase ticket.  
 returns as cancelled if not accepted  
 otherwise leaves cost the way it is*

orderTicket(userName):  
 *the main component of the program  
 passes information in Lists to check handle - ensures information is expected*

*gets information on their choices*

*returns individual ticket cost*

*main():*

asks user what they would like to do, exit, information or order a ticket then executes appropriate commands

once ticket(s) are ordered exiting main will print the tickets in ascending order and the total of the tickets

**Function: greeting()**

|  |  |  |
| --- | --- | --- |
| Input | Processing | Output |
| Prompt | Get user name | Print welcome message using username  userName |

Function greeting():

Display prompt (please enter your name)

userName = input

print “welcome userName”

return userName

**Function getAge():**

|  |  |  |
| --- | --- | --- |
| **Input** | **Processing** | **Output** |
| Prompt | Get age, ensure age is a number and with range 0 -130 | passengerAge |

**continueSwitch loop:**

Loop variable – continueSwitch

Exit condition – True - criteria has been met (input is number and between 0-130)

**Apply isdigit decision:**

|  |  |  |
| --- | --- | --- |
| **Path** | **Condition** | **Action** |
| Digit | Input is digit | Convert input to integer |
| Not digit | else | Print error message  Continue (request new input) |

**Apply number range decision:**

|  |  |  |
| --- | --- | --- |
| **Path** | **Condition** | **Action** |
| Expected range | Input within 0-130 | continueSwitch = True |
| Not in expected range | else | Print error message |

Function getAge():

continueSwitch = False

while continueSwitch = False

prompt - what is the age of the passenger?)

passengerAge = input

if passengerAge is not a digit

display error (Error: please input numbers)

continue

else

passengerAge = integer

if passengerAge between 0 and 130

continueSwitch = True

else

display error (Please ensure you entered the correct age of the passenger)

return passengerAge

**function checkHandle(prompt,codeList,word** **List ():**

|  |  |  |
| --- | --- | --- |
| **Input** | **Processing** | **Output** |
| prompt  codeList  wordList  input | Display prompt  Get input from user  Check that input matches codeList  Convert input into relevant word from word array | instanceInput |

**continueSwitch while loop:**

Loop variable – continueSwitch

Exit condition – True

**codeList for loop:**

Loop variable – i

Exit condition – i is greater then length of codeList

**Apply instanceInput decision**

|  |  |  |
| --- | --- | --- |
| **Path** | **Condition** | **Action** |
| Match | codeList[i] same as instanceInput | Replace instance input code with a word from wordList  continueSwitch = True  wrongCheck = False |
| Special case | length codeList same as 2 and instanceInput same as s | display special prompt – please enter the passenger’s name  instanceInput = input  continueSwitch = True  wrongCheck = False |

**Apply wrongCheck decision**

|  |  |  |
| --- | --- | --- |
| **Path** | **Condition** | **Action** |
| True | wrongCheck = True | Display error message - Error input not recognised. Please select what you would like by entering the corresponding letter in brackets |

**Algorithm:**

function checkHandle(prompt,codeList,wordList ():

continueSwitch = False

wrongCheck = True

while continueSwitch = False

display prompt – (prompt is passed in and has instructions relevant to the instance)

instanceInput = input

lower instance input characters

for i in range (0 to length of codeList)

if codeList [i] same as instanceInput

continueSwitch = True

instanceInput = wordList [i]

wrongCheck = False

else if length codeList same as 2 and instanceInput same as “s”

display special prompt – please enter the passenger’s name

instanceInput = input

continueSwitch = True

wrongCheck = False

if wrongCheck same as True

display error message - Error input not recognised. Please select what you would like by entering the corresponding letter in brackets

return instanceInput

**function costCalculation(tripDestination, flightType, seatType, seatClass, passengerAge):**

|  |  |  |
| --- | --- | --- |
| **Input** | **Processing** | **Output** |
| tripDestination  flightType  seatType  seatClass  passengerAge | If statements – assigning a value to each variable dependent on value  Determine cost of ticket by adding up values assigned  Then format values for later display | tripValue, classValue, typeValue, ageDiscountEligibility, cost |

**Apply tripDestination decision**

|  |  |  |
| --- | --- | --- |
| **Path** | **Condition** | **Action** |
| Cairns Oneway | TripDestination same as “Cairns”  flightType same as “One-way” | tripValue = 250 |
| Cairns Return | TripDestination same as “Cairns”  flightType same as “Return” | tripValue = 400 |
| Sydney Oneway | TripDestination same as “Sydney”  flightType same as “One-way” | tripValue = 420 |
| Sydney Return | TripDestination same as “Sydney”  flightType same as “One-way” | tripValue = 575 |
| Perth One way | TripDestination same as “Perth”  flightType same as “One-way” | tripValue = 510 |
| Perth Return | Else | tripValue = 700 |

**Apply seatClass decision**

|  |  |  |
| --- | --- | --- |
| **Path** | **Condition** | **Action** |
| Business | seatClass same as “Business” | classValue = 275 |
| Economy | seatClass same as “Economy” | classValue = 25 |
| Frugal | Else | classValue = 0 |

**Apply seatType decision**

|  |  |  |
| --- | --- | --- |
| **Path** | **Condition** | **Action** |
| Window | typeValue same as “Window” | typeValue = 75 |
| Aisle | typeValue same as “Aisle” | typeValue = 50 |
| Middle | Else | typeValue = -25 |

**Apply ageDiscount decision**

|  |  |  |
| --- | --- | --- |
| **Path** | **Condition** | **Action** |
| Infant | passengerAge less then or equal to 2 | ageDiscount = 0  ageDiscountEligibility = "(Eligible for infant ticket - free)" |
| Child | passengerAge less then or equal to 16 and greater then or equal to 3 | ageDiscount = 0.5  ageDiscountEligibility = “(Eligible for child ticket – half price)” |
| Normal | Else | ageDiscount = 1  ageDiscountEligibility = "(Not Eligible for child ticket)" |

**Algorithm:**

function costCalculation(tripDestination, flightType, seatType, seatClass, passengerAge):

if tripDestination same as "Cairns" and flightType same as "One-way"

tripValue = 250

else if tripDestination same as "Cairns" and flightType same as "Return"

tripValue = 400

else if tripDestination same as "Sydney" and flightType same as "One-way"

tripValue = 420

else if tripDestination same as "Sydney" and flightType same as "Return"

tripValue = 575

else if tripDestination same as "Perth" and flightType same as "One-way"

tripValue = 510

else:

tripValue = 700

if seatClass same as "Business"

classValue = 275

else if seatClass same as "Economy"

classValue = 25

else:

classValue = 0

if seatType same as "Window"

typeValue = 75

else if seatType same as "Aisle"

typeValue = 50

else:

seatType same as "Middle"

typeValue = -25

if passengerAge <= 2

ageDiscount = 0

ageDiscountEligibility = "(Eligiable for infant ticket - free)"

else if passengerAge <= 16 and passengerAge >= 3

ageDiscount = 0.5

ageDiscountEligibility = "(Eligiable for child ticket - half price)"

else:

ageDiscount = 1

ageDiscountEligibility = "(Not eligiable for child ticket)"

cost = (tripValue + classValue + typeValue) \* ageDiscount

format tripValue to $0.00

format classValue to $0.00

format typeValue to $0.00

return tripValue, classValue, typeValue, ageDiscountEligibility, cost

**function acceptPurchase(cost):**

|  |  |  |
| --- | --- | --- |
| **Input** | **Processing** | **Output** |
| Cost  input | Display prompt  Get input from user  If user enters yes nothing changes  Else if user enters no – cancel order | cost |

**acceptSwitch while loop:**

Loop variable – acceptSwitch

Exit condition – True

**Apply acceptPurchase decision**

|  |  |  |
| --- | --- | --- |
| **Path** | **Condition** | **Action** |
| Accepted | acceptPurchase same as "n" | Display – Your ticket has been purchased. Thank you for flying with Tropical Airlines. You have been returned to the main menu.  acceptSwitch = True |
| Denied | acceptPurchase same as "y" | cost = “CANCELLED ORDER”  Display - Your ticket has not been purchased. You have been returned to the main menu.  acceptSwitch = True |
| Error | else | Display – Error: Input not recognized. Please enter Y for yes or N for No. |

**Algorithm:**

**function acceptPurchase(cost):**

acceptSwitch = False

while acceptSwitch same as False

acceptPurchase = input("Would you like to purchase this ticket: (Y)es or (N)o")

acceptPurchase = acceptPurchase.lower()

if acceptPurchase same as "y"

display – “Your ticket has been purchased. Thank you for flying with Tropical Airlines. You have been returned to the main menu.”

acceptSwitch = True

else if acceptPurchase same as "n"

cost = "CANCELLED ORDER"

display - "Your ticket has not been purchased. You have been returned to the main menu."

acceptSwitch = True

else:

display - "ERROR: Input not recognized. Please enter Y for yes or N for No.”

return cost

**function ordetTicket(userName):**

|  |  |  |
| --- | --- | --- |
| **Input** | **Processing** | **Output** |
| username | Assign codeList, wordlist and prompt as required.  Frugal cannot choose seats – assign middle.  Calculate cost  Ensure they want ticket | cost |

**Apply acceptPurchase decision**

|  |  |  |
| --- | --- | --- |
| **Path** | **Condition** | **Action** |
| Frugal Class | seatClass same as Frugal | Assign seatType “Middle” |
| Not frugal class | else | Assign seatclass = Call checkHandle |

**Algorithm:**

**def orderTicket(userName):**

codeList = ["y", "s"]

wordList = [userName]

prompt = "Is the ticket for (Y)ou or (S)omeone else?"

passengerName = checkHandle(prompt,codeList,wordList)

passengerAge = getAge()

codeList = ["c", "s", "p"]

wordList = ["Cairns","Sydney","Perth"]

prompt = "Please select the destination for your return trip. Base fare prices are listed below: \n(C)airns – $400 \n(S)ydney – $575 \n(P)erth - $700"

tripDestination= checkHandle(prompt,codeList,wordList)

codeList = ["o", "r"]

wordList = ["One-way","Return"]

prompt = "Is this a: \n(R)eturn trip \n(O)ne-Way"

flightType = checkHandle(prompt,codeList,wordList)

codeList = ["b", "e","f"]

wordList = ["Business", "Economy", "Frugal"]

prompt = "Please choose the type of fare. Fees are displayed below and are in addition to the basic fare. \nPlease note choosing Frugal fare means you will not be offered a seat choice. \n(B)usiness - $275 \n(E)conomy - $25 \n(F)rugal - $0"

seatClass = checkHandle(prompt, codeList, wordList)

if seatClass == "Frugal":

seatType = "Middle"

else:

codeList = ["w", "a","m"]

wordList = ["Window", "Aisle", "Middle"]

prompt = "Please choose the seat type. Choosing the middle seat will deduct 25 from the total fare. \n(W)indow $75 \n(A)isle $50 \n(M)iddle -$25"

seatType = checkHandle(prompt, codeList, wordList)

#calculate cost from codes

tripValue, classValue, typeValue, ageDiscountEligibility, cost = costCalculation(tripDestination,flightType,seatType,seatClass,passengerAge)

costDisplay = "${:,.2f}".format(cost)

#ensure user wants ticket

Display -" Ticket for: " + passengerName + "Passenger age: " + passengerAge + ageDiscountEligibility + "Trip destination: " + tripDestination + "(" + flightType + ") " + tripValue + " Seat class: " + seatClass + " " + classValue + " Seat type: " + seatType + " " + typeValue + " Total cost: " + costDisplay

cost = acceptPurchase(cost)

return cost

**main():**

|  |  |  |
| --- | --- | --- |
| **Input** | **Processing** | **Output** |
| input | Determine what object user wants – information, order or menu.  Information = get information  Order = call order function, append value to end of list  Exit = determine how long orders list is, display in appropriate format  Length 0 = no tickets  Length 1 = one ticket, total will be same as that ticket  Length more then 1 = number of tickets, use for loop to dispay each order. will need for loop to calculate total. | Informational message  Summary of orders |

**acceptSwitch while loop:**

Loop variable – exitProgram

Exit condition – True

**Apply userInput decision**

|  |  |  |
| --- | --- | --- |
| **Path** | **Condition** | **Action** |
| Information | userInput == "i": | Display - informational message |
| Order Ticket | userInput == "o": |  |
| Exit | userInput == "e": |  |
| Error | else |  |

**Algorithm:**

**def main():**

userName = greeting()  
 exitProgram = False  
 set ticketCost list  
  
 while not exitProgram:  
 display - Tropic Airlines Ticket Ordering System: (I)nstructions, (O)rder ticket, (E)xit   
 userInput = input  
 userInput = make lower case  
 if userInput same as "o"  
 ticketInformation = orderTicket(userName)  
 if ticketInformation same as "CANCELLED ORDER"  
 display -The ticket was not processed.  
 else:  
 ticketInformation attach to end of ticketCost list  
  
 else if userInput same as "e"  
 ticketCost sort in ascending order   
  
 detemine how long the array is, depending on length different actions  
 if ticketCostLength same as 1  
 ticketCost[0] = format to $0.00  
 display - userName + ", your order is: " + ticketCost[0] + " Your final total is: " + ticketCost[0]  
 else if ticketCostLength > 1:  
 display - userName + " your orders are: "  
 for i in range(0, len(ticketCost)):  
 displayCost = format to $0.00  
 display - "Ticket " + (counter plus 1 + displayCost)  
  
 total=0  
 for price in ticketCost:  
 total = total + price  
 displayTotalCost = format to $0.00  
 display - "Your final total is: " + displayTotalCost  
  
  
 display – “Thank you for choosing Tropical Airlines for your air travel needs.”  
 exitProgram = True  
  
 else if userInput same as "i"  
 display - "Thank you for choosing Tropical Airlines for your air travel needs. You will be asked questions regarding what type of ticket you would like to purchase as well as destination information. We also offer 50% discounted fares for children under the age of 16. Infants under the age of 2 are free."  
 display - "You have been returned to the main menu."  
 else:  
 display - "ERROR: Input not recognised. Please enter the letter in brackets to complete the intended action."  
  
main()