```
# **   DECLARE CONSTANTS

# These are the options of the various properties of the pizza available
sizesAvailable = ["Small", "Medium", "Large"]  # The size of the pizza
basesAvailable = ["Thick", "Thin"]  # The type of base of the pizza
toppingsAvailable = ["Pepperoni", "Chicken", "Extra Cheese", "Mushrooms", "Spinach",
"Olives"]  # The toppings available

maxToppings = 3  # The maximum number of toppings that can be taken

# **   DECLARE VARIABLES
CurrentID = 0  # The running unique ID of the order
ordersCount = 0  # The running total of the number of confirmed orders
close = False  # status of more orders

highest = 0.0
highestIndex = 0
lowest = 1000.0
lowestIndex = 0
toppingsSum = 0.0

orderData = []  # Running tracker of all the items of one order

# Initialize the array with all values 0
totalSizes = [0, 0, 0]  # Set values for 3 sizes
totalBases = [0, 0]  # Set values for 2 bases
totalToppings = [0, 0, 0, 0, 0, 0]  # Set values for 6 toppings

# **   TASK 1
# Use a default status "Alter" to customize the pizza
# Input the values of each attribute for count in range( validate them
# Give the customer a choice to alter the order, confirm it OR cancel it
# If they choose to alter, re-input the values
# If they confirm it, provide them with a new order number.

# **   TASK 2
# Increment a counter of number of pizzas if an order is confirmed
# Add the value of the Counters[] to the TotalCounters[]
# Output the number of pizzas ordered.

while (close != True):

    status = "Alter"  # Default status to input values

    # Input for count in range( validate the values
    while status == "Alter":  # As long as the status is "Alter"

        # Reset the running tracker
        orderData = []  # Initialize to have 0 toppings

        # Output the available options

        # Output the sizes
        print "\nThe following sizes are available to choose from:"
        for count in range(3):  # Iterate 3 times for 3 sizes
            print sizesAvailable[count] + ',',  # Output the available sizes

        # Output the bases
        print "\n\nThe following bases are available to choose from:"
        for count in range(2):  # Iterate 2 times for 2 pizza bases
            print basesAvailable[count] + ',',  # Output the available bases

        # Output the toppings
        print "\n\nThe following toppings are available to choose from:"
        for count in range(6):  # Iterate 6 times for 6 toppings
            print toppingsAvailable[count] + ',',  # Output the available toppings

        size = ""  # Enable the while loop to run by making the size invalid

```

```
69          # Input and validate the size of the pizza
70          while (size != "Small") and (size != "Medium") and (size != "Large"):  #
            Validation loop
71              size = raw_input("\n\nPlease enter the size of the pizza you would like:
                ")  # Input the size
72
73              if (size != "Small") and (size != "Medium") and (size != "Large"):  # If
                the size is invalid
74                  print "The size you have entered is invalid. Please re-enter the
                    size from one of the options above."  # Print error message and ask
                    for correction
75
76          # Unless the size is invalid, break out of the loop
77
78          # Input and validate the base of the pizza
79
80          base = ""  # Enable the while loop to run by making the base invalid
81
82          while (base != "Thick") and (base != "Thin"):  # Validation loop
83              base = raw_input("\nPlease enter the pizza base you would like: ")  #
                Input the base
84
85              if (base != "Thick") and (base != "Thin"):  # If the base is invalid
86                  print "The base you have entered is invalid. Please re-enter the
                    base from one of the options above."  # Print error message and ask
                    for correction
87
88          # Unless the base is invalid, break out of the loop
89
90          # Input and validate the number of toppings the customer wants
91          print   # Input prompt
92
93          toppingChoice = 100  # Enable the while loop to run by making the number of
            toppings invalid
94
95          while not ((toppingChoice <= 3) and (toppingChoice >= 0)):  # Validation loop
96              toppingChoice = int(input("How many toppings do you want on your pizza?
                You may enter any whole number between 0 and 3: "))  # Input the number
                of toppings the user wants
97
98              if not ((toppingChoice <= 3) and (toppingChoice >= 0)): # If the number
                of toppings is invalid
99                  print "You have entered an invalid number of toppings. Please
                    re-enter any whole number between 0 and 3."  # Throw error message
                    and ask for correction
100
101         # Unless the number of toppings is greater than 3, break out of the loop
102
103         numberOfItems = 3 + toppingChoice  # Calculate the total number of items
            based on the number of toppings
104
105         orderData = range(numberOfItems)  # Declare an array with as many elements
            as in the order
106
107         # Store the data acquired so far
108         orderData[0] = size  # Store the size
109         orderData[1] = base  # Store the base
110         orderData[2] = numberOfItems  # Store the total number of items
111
112         for outsideCount in range(toppingChoice):  # Iterate as many times as the
            toppings taken
113
114             # Input for count in and validate the topping of the pizza
115
116             topping = ""  # Enable the while loop to run by making the topping invalid
117
118
119
120
```

```
121                 while (topping != "Pepperoni") and (topping != "Chicken") and (topping !=
                     "Extra Cheese") and (topping != "Mushrooms") and (topping != "Spinach")
                    and (topping != "Olives"):  # Validation loop
122                     topping = raw_input("Please enter topping " + str(outsideCount + 1) +
                         " of the pizza you would like: ")  # Input the topping
123
124                     if (topping != "Pepperoni") and (topping != "Chicken") and (topping
                        != "Extra Cheese") and (topping != "Mushrooms") and (topping !=
                        "Spinach") and (topping != "Olives"):  # If the topping is invalid
125                         print "The topping you have entered is invalid. Please re-enter
                            the topping from one of the options above."  # Print error
                            message and ask for correction
126
127                 # Unless the topping is invalid, break out of the loop
128
129                 orderData[2 + outsideCount] = topping  # Store the validated topping in
                    the array
130
131             # Move on to the next topping
132
133             status = raw_input("\nDo you want to Alter your order, Confirm or Not
                proceed? ")  # Input whether the customer wants to alter their order,
                confirm it or cancel it
134
135         # Unless they want to alter their order, break out of the loop
136
137         # Give the customer a unique order ID if they have confirmed it
138         if status == "Confirm":  # If the customer has confirmed their order
139
140             print "\nYour unique order number is: ", CurrentID  # Print out the unique ID
141             CurrentID = CurrentID + 1  # Increment the ID for the next confirmed order
142             ordersCount = ordersCount + 1  # Increment the counter for confirmed orders
143
144             # Record how many of each size has been ordered
145             for count in range(3):  # Iterate 3 times for 3 sizes
146                 if orderData[0] == sizesAvailable[count]:  # If a size is recorded
147                     totalSizes[count] = totalSizes[count] + 1  # Increment the counter
148
149             # Record how many of each pizza base has been ordered
150             for count in range(2):  # Iterate 2 times for 2 pizza bases
151                 if orderData[1] == basesAvailable[count]:  # If a pizza base is recorded
152                     totalBases[count] = totalBases[count] + 1  # Increment the counter
153
154             # Record how many of each topping has been ordered
155             for outsideCount in range(toppingChoice):  # Run as many times as the number
                of toppings taken
156                 for insideCount in range(6):  # Iterate 6 times for 6 toppings
157                     if orderData[2 + outsideCount] == toppingsAvailable[insideCount]:  #
                        If a topping has been ordered
158                         totalToppings[insideCount] = totalToppings[insideCount] + 1  #
                            Increment the counter
159
160         close = input("\nDo you want to exit the program? ")  # Ask the staff if all
            orders are done
161
162     # Break out of the loop unless more pizzas are to be ordered
163
164     print "\n\n", ordersCount, "pizzas were ordered."  # Output how many pizzas were
        ordered
165
166     # **   TASK 3
167     # Calculate the total number of toppings ordered
168     # Calculate the highest ordered toppings
169     # Calculate the lowest ordered toppings
170     # Express both values as a percentage of the total orders
171
172     for count in range(6):  # Iterate 6 times for 6 toppings
173         toppingsSum = toppingsSum + totalToppings[count]  # Add to the running total to
            calculate the sum
```

```
174         # Calculate the highest sales
175         if totalToppings[count] > highest:  # If the current topping sold more than the
            running most popular topping
176             highest = totalToppings[count]  # Update the running most popular topping
177             highestIndex = count  # Record the array index of the topping
178 
179         # Calculate the lowest sales
180         if (totalToppings[count] < lowest) and (totalToppings[count] > 0):  # If the
            current topping sold less than the running least popular topping for count in
            range( it sold in the first place
181             lowest = totalToppings[count]  # Update the running least popular topping
182             lowestIndex = count  # Record the array index of the topping
183 
184     print toppingsAvailable[highestIndex], "was the most popular topping and accounted
        for", ((highest/toppingsSum) * 100), "% of the toppings sales."  # Output the most
        popular toppings
185     print toppingsAvailable[lowestIndex], "was the least popular topping and accounted
        for", ((lowest/toppingsSum) * 100), "% of the toppings sales."  # Output the least
        popular toppings
186 
187     # This is the end of the program
188     # All required tasks have been completed.
189 
```