CAT 2 Take Away

SIT 263 Software Engineering

Instructions:

Everyone should attempt this assignment. Use your notes and other sources if necessary. The answers should be typed and printed on paper and submitted after ONE week. Each question carries 10 marks making a total of 30 marks. Remember to write your names and registration numbers on the answer book

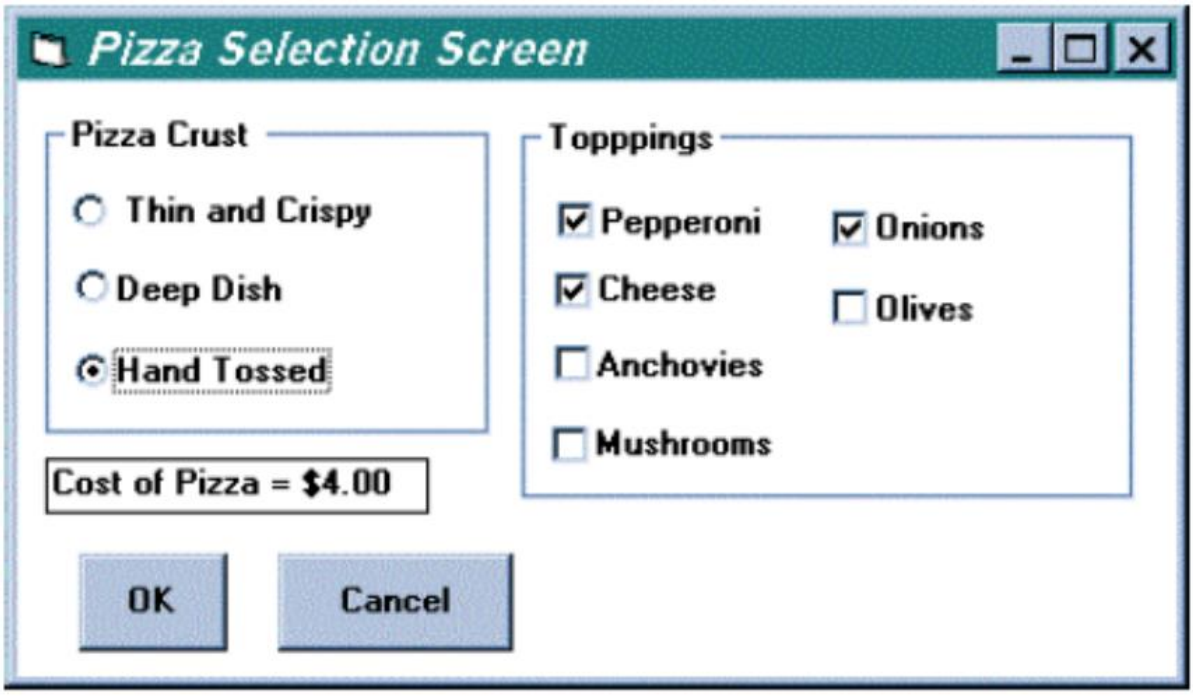
Question ONE

Function Point Analysis

You are about to develop an app for a Pizzeria. People can order pizzas with the app. You have started with the GUI design.

The following shows two options for the Pizza Selection Screen.

Option 1:



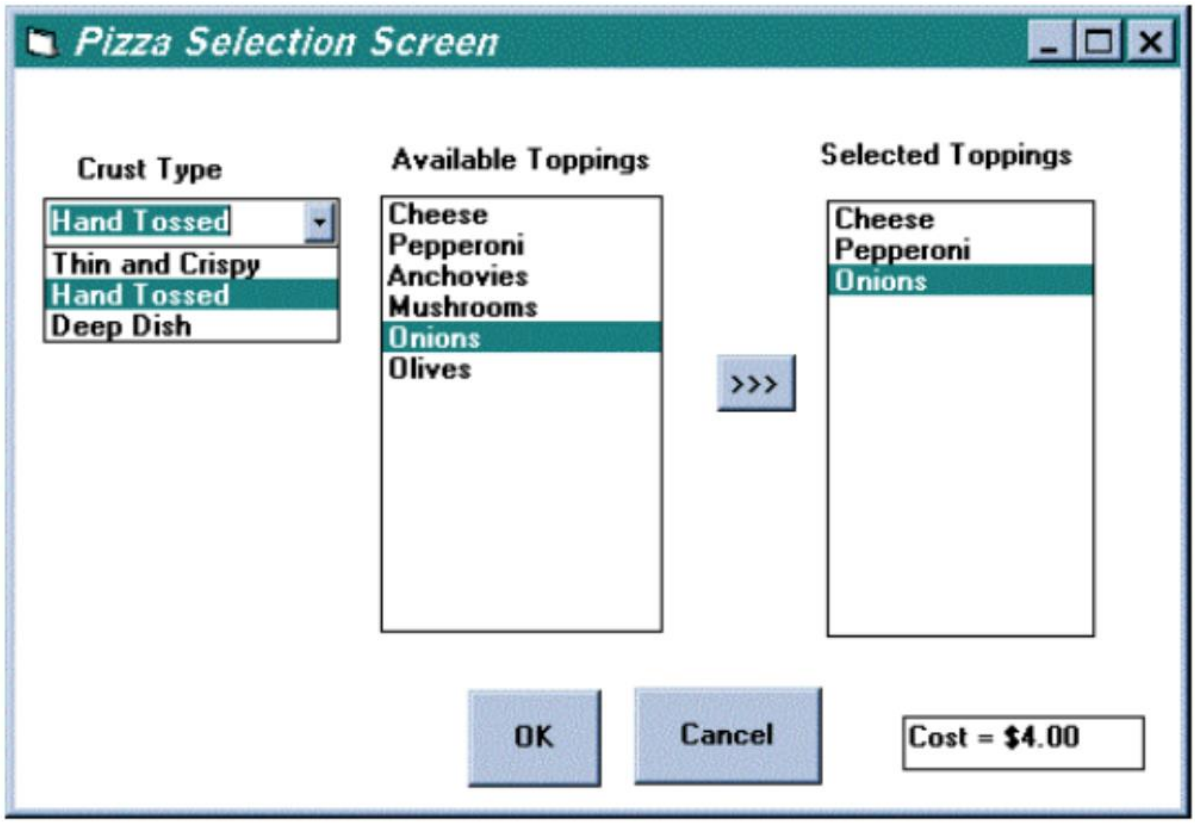
Explanation:

• Toppings are read from another application (kitchen application). If the topping is not available it is not displayed.

• The cost of the Pizza is calculated automatically.

• When the OK button is clicked the Toppings, Pizza Crust Type and Cost of Pizza are saved for further processing.

Option 2:



Explanation:

• The Items in the drop down box are hard coded – not read from a file.

• Available Toppings are read from another application (kitchen application).

• When a Topping is selected from Available Toppings it is copied to Selected Toppings

• The Cost of the Pizza is automatically calculated.

• When the OK button is clicked the Selected Toppings, Pizza Crust Type and Cost of Pizza are saved for further processing.

To Do:

1. What are the function types (EI, EQ, EO, EIF, and ILF) and elements (DET, FTR, RET) in each of the options? Are there differences?

2. What is the unadjusted function point count for the Pizza Selection Screen Option 2?

3. Assume total number of unadjusted function point for the Pizza App equals 1000. You have the following information about system characteristics:

- No special performance requirements were stated

- 50% of transactions are interactive data entry

What is the total number of adjusted function points of the system?

How would this influence the FP count?

Question TWO

The following question is based on **Halstead's Software Metrics**

Below are TWO program fragments, one written in the C++ programming language and the other written in the Python programming language. Study the program fragments and answer questions below it.

```
int main()
{
  int n, num, sum = 0,digit;
  cout << "Enter a positive  integer: ";
  cin >> n;
  num = n;
  while (num != 0)
  {
     digit = num % 10;
     sum += digit * digit * digit;
     num /= 10;
  }
  if (sum == n)
    cout << n << " is an Armstrong number.";
  else
    cout << n << " is not an Armstrong number.";
  return 0;
}
```

(b) Sample Code 2 in Python:

```
num = int(input("Enter a number: "))
```

```
sum = 0
temp = num
while temp > 0:
   digit = temp % 10
   sum += digit ** 3
   temp /= 10
if num == sum:
   print(num,"is an Armstrong number")
else
   print(num,"is not an Armstrong number")
```

To do:

Compare the TWO Program fragments in terms of Halsted's Metrics:

a) Measure N (Length of the program):
b) Measure n (Vocabulary of the program):
c) Measure V (Volume of the program
d) Measure D (Difficulty of the program):
e) Measure V* (Potential or minimal Volume V*):
f) Measure L (Implementation Level):
g) Measure L' (Program Level Estimator L'):
h) Measure I (Intelligent Content):
i) Measure E - Effort required implementing or understanding the programe.
j) Measure B - Number of bugs expected in the program: Is proportional to the effort? B=E0.667/3000
k) Measure T -Estimated time is taken to write the program:

Question THREE

1) A given project has 3 user inputs, 5 user outputs, 2 inquiries, 2 files, and 2 external interfaces. All of these are average complexity EXCEPT 2 of the inputs are complex, two of the outputs are complex, and one of the outputs is simple. Adjustment factors are all moderate except that the system will require a significant amount of online data entry, and it is essential that the code is designed with reuse in mind. Calculate the number of Function Points for this system. (Show all your workings).