Usman Institute of Technology

Affiliated with NED University of Engineering and Technology

Spring 2020

CS121Object Oriented Programming

Instructor: Dr. Lubaid Ahmed

Class: UIT SP20 19B-SE Sec B

Midterm Exam -Assignment 2(Max. Marks: 10)

**Name: SAIFULLAH Course Code: CS-121 Department: Computer Science**

**Roll #: 18B-092-CS Course Title: Object Oriented Prog Batch: 2018**

**Section: SE-B Instructor: Dr. Lubaid Ahmed Year: Spring-2020**

**Submission Date: Saturday, July 25, 2020**

Question 2: Based on the project in assignment 1write how many algorithms your project may use. Identify them and write them in proper pseudo code formatting.

1. First you identify the algorithms.

* There are 3 algorithms used in my project.
  + Assign\_id() method in (sale, purchase, expense, stock class)
  + Total method in (sale, purchase, expense, stock class)
  + Profit method in cashflow class

1. Write steps of each algorithm.

* Assign\_id() steps:

1. Define new method assign inside assign\_id method()
2. Convert the default string id into list data type.
3. Add the list elements into a variable using indexing.
4. Change the datatype of variable to integer.
5. Add 1 to variable.
6. Then check length of variable with datatype string if length is 1 update variable by concatenating two zeros with variable type string (var= ‘0’+’0+’str(var))
7. if length is 2 update variable by concatenating one zero with variable type string (var= ’0+’str(var))
8. if length is 3 update variable to string datatype.
9. Then update default id with variable
10. Return variable
11. Then outside of assign() function and inside in assign\_id() method
12. Check in database there is any entry or not
13. If entry found fetch the last entry id and set id to default string id and call assign function.
14. Else if no entry in database just call assign function

* Total() steps:

1. Fetch the amounts of all record from database and store in variable
2. Then use two for loops to take out the amounts from list and tiples
3. Make another variable before loops to add all amounts in that variable within the 2nd loop.
4. Return the variable which stored all added amount outside the loop.

* Profit() steps:

1. Get sales total from sales class and store in 1st variable.
2. Get purchase total from purchase class and store in 2nd variable
3. Get expense total from Expense class and store in 3rd variable
4. Add purchase total and expense total and store in 4th variable.
5. Now from Sales total less purchase total and expense total(4th variable) and store in 5th variable.
6. Return this 5th variable.
7. Make pseudo code.

* Assign\_id()
  + Start
  + Define assign\_id method
  + Define assign function inside assign\_id method
  + Set variable a to default string id with changing da type to list
  + Set variable b to list index 1-3
  + Set the datatype of variable b t integer
  + If length of string b equals to 1
  + Update variable b with two 0’s string datatype with variable b string datatype
  + Elif length of string b equals to 2
  + Update variable b with one 0 string datatype with variable b string datatype
  + Elif length of string b equals to 3
  + Update variable b with b variable datatype string
  + Set new variable d to list 0th element and string b variable
  + Set the default id to d variable
  + Return d variable inside assign function
  + Then in assign\_id method checks in database select ids from table and store in variable a
  + Set variable k to none
  + Loop the variable a
  + Again, loop the first loop
  + Update variable k to last loop value
  + If k not equal to none set default string id to variable k
  + And then call assign function and store in variable i
  + Return i
  + Else just call assign id function store in new variable j
  + Return j
  + End
  + Python code:
  + def assign\_id(self):
  + def assign():
  + a=list(Sales.sale\_id)
  + b=a[1]+a[2]+a[3]
  + b=int(b)
  + b+=1
  + if len(str(b))==1:
  + b='0'+'0'+str(b)
  + elif len(str(b))==2:
  + b='0'+str(b)
  + elif len(str(b))==3:
  + b=str(b)
  + d=a[0]+str(b)
  + Sales.sale\_id=d
  + return d
  + a=cursor.execute("Select ID from Sales")
  + con.commit()
  + k= None
  + for i in a:
  + for j in i:
  + k=j
  + if k!= None:
  + Sales.sale\_id=k
  + i= assign()
  + return i
  + else:
  + j= assign()
  + return j
* Total()
* Start
* Fetch amount from database and store in variable
* Make new variable set value to 0
* Loop the fetch variable to take out tuple from list
* Again, loop to take out the single value from tuple
* Update the 0 variable by adding the values of amount
* Return the update variable
* End
* Python code:
* sat = cursor.execute("SELECT Amount from Sales")
* st = 0
* for i in sat:
* for a in i:
* st += a
* return st
* Profit()
* Start
* Get Purchase total value and store in variable a
* Get Expense total value and store in variable b
* Get Sales total value and store in variable c
* Set the variable datatype to integer
* Make new variable ex and add variable a and b. (a+b)
* Make profit variable and subtract ex from variable c
* Return profit variable
* End
* Python code:
* a=Purchase.total(self)
* int(a)
* b=Expense.total(self)
* int(b)
* c= Sales.total(self)
* int(c)
* ex=b+a
* profit = c-ex
* return profit

1. According to SQA (Software Quality Assurance Department) your code is missing with Exceptional Handling, so any time when there is an error the project has stopped and open debugger mode. Use Error Exceptional handling technique and methods (29 different error exceptional handling types have been covered in lecture) protect your code for being open in debugger mode.

I have tried to use exceptional handling to prevent project from opening in debugger mode but it’s not 100% accurate some exceptions are still missing.