Problem 1:

* Read the XML file.
* Add new or delete existing employees.
* Write to the XML file.

Code snippet and screenshot:

Read the XML file:

def read\_xml\_file(self, file\_path):  
 *"""  
 in:  
 file\_path: path of xml file for reading.  
  
 out:  
 return the parent object of xml.  
  
 """* try:  
 tree = ET.parse(file\_path)  
 root = tree.getroot()  
 return root  
 except Exception as ex:  
 print("No xml file at given file path location: {}".format(ex))

write to the XML file:

def write\_xml\_file(self, root, file\_path):  
 *"""  
 in:  
 root: parent node/object of xml.  
 file\_path: path of xml file for reading.  
  
 out:  
 write the complete to xml file at given file\_path  
  
 """* try:  
 with open(file\_path, 'wb') as f:  
 f.write(ET.tostring(root))  
 f.close()  
 except Exception as ex:  
 print("Error file writing to xml file. {}".format(ex))

Add new employee

def add\_employee(self, root, attributes\_list, value\_list, file\_path):  
 *"""  
 in:  
 root: parent node/object of xml.  
 attributes\_list: List of attribute for new employee  
 value\_list: List of value corresponding to each attributes  
 file\_path: save the xml to given file\_path having new employee details.  
  
 out:  
 added new employee to existing xml file with given employee details.  
  
 """* try:  
 attributes\_obj = []  
 new\_emp = Element("employee")  
  
 for attribute in attributes\_list:  
 attr = ET.SubElement(new\_emp, attribute)  
 attributes\_obj.append(attr)  
  
 for i in range(len(attributes\_obj)):  
 attributes\_obj[i].text = value\_list[i]  
  
 root.append(new\_emp)  
  
 self.write\_xml\_file(root, file\_path)  
 except Exception as ex:  
 print(ex)

function call:

xml\_processor1 = XmlProcessor()  
filepath = r'C:\Users\q1036048\Desktop\test\_peer.xml'  
parent\_obj = xml\_processor1.read\_xml\_file(filepath)  
  
# add new employee  
attributes\_list = ['name', 'age', 'designation']  
value\_list = ['Ram', '30', 'Software Engineer']  
xml\_processor1.add\_employee(parent\_obj, attributes\_list, value\_list, filepath)

Screenshot:

Ram is added in employee list.



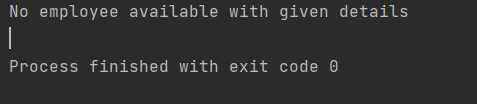
Delete new Employee:

def delete\_employee(self, root, file\_path, \*\*kwargs):  
 *"""  
 in:  
 root: parent node/object of xml.  
 file\_path: save the modified xml to given file\_path.  
 \*\*kwargs: can pass multiple keywords arguments e.g, name='Mohan', age='25'.  
  
 out:  
 Based on argument provide, if details are present then that employee details will get removed from xml file.  
  
 """* is\_available = False  
 try:  
 for key, value in kwargs.items():  
 emp\_list = root.findall('employee')  
 for emp in emp\_list:  
 for node in emp.findall(key):  
 if node.text.lower() == value.lower():  
 is\_available = True  
 root.remove(emp)  
  
 if not is\_available:  
 print("No employee available with given details")  
  
 self.write\_xml\_file(root, file\_path)  
 except Exception as ex:  
 print(ex)

function call:

xml\_processor1 = XmlProcessor()  
filepath = r'C:\Users\q1036048\Desktop\test\_peer.xml'  
parent\_obj = xml\_processor1.read\_xml\_file(filepath)  
  
#delete employee  
xml\_processor1.delete\_employee(parent\_obj, filepath, name='Meera')

screenshot:



New function call:

#delete employee  
xml\_processor1.delete\_employee(parent\_obj, filepath, name='Ram')

Screenshot:



Ram is deleted from employees list.

Problem 2:

Please extend the code to include a new address node and create the provision to add new or delete existing employees.

def add\_sub\_attribute\_for\_employee(self, root, attribute, sub\_attributes\_list, value\_list, file\_path, \*\*kwargs):  
 *"""  
 in:  
 root: parent node/object of xml.  
 attribute: attribute name for employee i.e, address, Qualification  
 attributes\_list: List of attribute for new employee i.e, for address (doorNo, street, State)  
 value\_list: List of value corresponding to each attributes i.e, for address (10, 'Edapally', 'Kerala')  
 file\_path: save the xml to given file\_path having new employee details.  
 kwargs: keyword arguments to decide for which employee new attributes is to be added.  
 out:  
 add new attributes and it's sub attributes for given employee to existing xml file.  
  
 """* try:  
 sub\_attributes\_elements = []  
 for key, value in kwargs.items():  
 emp\_list = root.findall('employee')  
 for emp in emp\_list:  
 for node in emp.findall(key):  
 if node.text.lower() == value.lower():  
  
 new\_attr = ET.SubElement(emp, attribute)  
  
 for attribute in sub\_attributes\_list:  
 sub\_attr = ET.SubElement(new\_attr, attribute)  
 sub\_attributes\_elements.append(sub\_attr)  
  
 for i in range(len(sub\_attributes\_elements)):  
 sub\_attributes\_elements[i].text = value\_list[i]  
  
 self.write\_xml\_file(root, file\_path)  
 except Exception as ex:  
 print(ex)

function call:

xml\_processor2 = XmlProcessor()  
filepath = r'C:\Users\q1036048\Desktop\test\_peer.xml'  
parent\_obj = xml\_processor2.read\_xml\_file(filepath)  
  
  
# # add attribute to existing employee  
attribute = 'address'  
sub\_attributes\_list = ['doorNo', 'street', 'town', 'state']  
value\_list = ['10', 'Edapally', 'Ernakulam', 'Kerala']  
xml\_processor2.add\_sub\_attribute\_for\_employee(parent\_obj, attribute, sub\_attributes\_list, value\_list, filepath, name='Ram')

Screenshot:



Address attribute has bee added to given Employee.

To add new node, just call with function with required details. Example to add qualification node:

attribute = 'Qualifications'  
sub\_attributes\_list = ['UG', 'PG', 'certifications']  
value\_list = ['CUSAT', 'CUSAY', 'CKAD']  
xml\_processor2.add\_sub\_attribute\_for\_employee(parent\_obj, attribute, sub\_attributes\_list, value\_list, filepath, name='Ram')

Screenshot:

