

Compiler Construction

Assignment 3



Roll no:	200901018
Name:	Muhammad Tayyab
Batch & Section:	BS-CS-01 (B)
Date of Submission:	14/01/2023

Overview of Parser

Module 1: Parse an XML file using python and put the data into an excel worksheet.

Parsing an XML file involves analysing the structure and content of an XML document and turning it into a more easily understandable format, such as a tree or a list of objects. This process is useful for extracting specific data from an XML file, as well as for manipulating and modifying the contents of the file. There are several tools and libraries available for parsing XML files, including DOM (Document Object Model) parsers, SAX (Simple API for XML) parsers, and XML Pull parsers. Each of these approaches has its own strengths and weaknesses, and the best choice will depend on the specific requirements of the task at hand. Regardless of the approach taken, proper handling of errors and edge cases is important to ensure that the XML file is parsed accurately and efficiently.

Solution Code:

```
import xml.etree.ElementTree as ElemT
import openpyxl

def main():
    # Parse the XML file
    tree = ElemT.parse('compiler.xml')

    # open Excel worksheet
    workbook = openpyxl.Workbook()
    sheet = workbook.active

    # Get the root element
    root = tree.getroot()

    # Assigning Column Names
    sheet.cell(1, 1).value = 'Book ID'
    sheet.cell(1, 2).value = 'Author Name'
    sheet.cell(1, 3).value = 'Title'
    sheet.cell(1, 4).value = 'Genre'
    sheet.cell(1, 5).value = 'Price'
    sheet.cell(1, 6).value = 'Publish date'
    sheet.cell(1, 7).value = 'Description'

    # Iterate over the children of the root element
    for i, child in enumerate(root):
        # Print the tag and text for each child element
        print(child.tag, child.attrib['id'])
        sheet.cell(row=i + 2, column=1).value = child.attrib['id']

        for j, subChild in enumerate(child):
            # Print the tag and text for each subChild element
            print(f"    {subChild.tag}: {subChild.text}")
```

```
        sheet.cell(row=i + 2, column=j+2).value = subChild.text

    workbook.save('200901018_Assign_03.xlsx')

if __name__ == '__main__':
    main()
```

Explanation:

This code reads data from an XML file called "compiler.xml" and writes the data to an Excel spreadsheet. It first uses the `xml.etree.ElementTree` library to parse the XML file, and then uses the `openpyxl` library to create and edit an Excel worksheet.

It assigns column names in the first row of the sheet, and then iterates over the elements in the XML file to write the data to the sheet, with each element's data being written to a new row.

Finally, it saves the Excel file in the same directory.

Code File:

The code file is attached alongside the pdf, it has also been uploaded to GitHub

<https://github.com/ProgradeX/Semester-Project-Archive/tree/master/Compiler%20Construction>