Assignment:-1

1. Create a simple XML file with a root element <student> and child elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<student>
    <name>John Doe</name>
    <age>21</age>
    <course>Computer Science</course>
</student>
```

2. Write an XML document to store book details (title, author, price).

```
<?xml version="1.0" encoding="UTF-8"?>
<book>
    <title>XML Basics</title>
    <author>Jane Smith</author>
    <price>299.99</price>
</book>
```

3. Define an XML declaration with version and encoding at the beginning of an XML file.

```
<?xml version="1.0" encoding="UTF-8"?>
```

4. Create an XML file with nested elements representing an employee and their department.

```
<?xml version="1.0" encoding="UTF-8"?>
<employee>
    <name>Alice Johnson</name>
    <department>
        <name>IT</name>
        <location>Building 1</location>
    </department>
</employee>
```

5. Demonstrate case sensitivity in XML by creating two similar tags with different cases.

```
<?xml version="1.0" encoding="UTF-8"?>
<example>
    <Name>Tom</Name>
    <name>Jerry</name>
</example>
```

6. Write an XML document with a document prolog and document element section.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Document Prolog Example -->
<document>
    <title>XML Sample</title>
</document>
```

7. Create an XML document with attributes instead of nested elements for storing product details.

```
<?xml version="1.0" encoding="UTF-8"?>
<product name="Laptop" price="75000" brand="Dell"/>
```

8. Define an XML element with both attributes and nested elements inside it.

```
<?xml version="1.0" encoding="UTF-8"?>
<employee id="101" role="Manager">
    <name>Samuel Green</name>
    <age>40</age>
</employee>
```

9. Write an XML file that follows proper syntax and structure rules.

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
    <to>Alex</to>
```

```xml
    <from>Sam</from>
    <message>Meeting at 3 PM</message>
</note>
```

10. Create an XML document for a list of students with their id, name, and age.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<students>
    <student id="1">
        <name>John</name>
        <age>20</age>
    </student>
    <student id="2">
        <name>Mary</name>
        <age>22</age>
    </student>
</students>
```

11. Demonstrate XML comments by adding a comment inside an XML document.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<note>
    <!-- This is a comment -->
    <to>Alice</to>
    <from>Bob</from>
</note>
```

12. Create an XML file for a company’s employee records with multiple employee entries.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<employees>
    <employee>
        <name>Emma</name>
        <position>Developer</position>
    </employee>
    <employee>
        <name>Ryan</name>
        <position>Designer</position>
    </employee>
</employees>
```

13. Write an XML document containing a root element <catalog> with multiple <book> entries.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<catalog>
    <book>
        <title>Learn XML</title>
        <author>Adam Ray</author>
    </book>
    <book>
        <title>Advanced XML</title>
        <author>Susan Lee</author>
    </book>
</catalog>
```

14. Validate an XML file to ensure it follows correct nesting and syntax rules.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<valid>
    <tag1>
        <tag2>Content</tag2>
    </tag1>
</valid>
```

15. Write an XML file with special characters and use CDATA to handle them.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<message>
```

```
    <![CDATA[This message contains special characters like <, >, &, etc.]]>
</message>
```