ASSIGNMENT-1

(a)

(i) What is XML?

XML stands for Extensible Markup Language. It is It is a text-based markup language derived from Standard Generalized Markup Language (SGML). XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

(ii) Why do we need DTD?

DTD stands for Document Type Definition/Declaration. The document type (DOCTYPE) declaration consists of an internal, or references an external Document Type Definition (DTD). It can also have a combination of both internal and external DTDs. The keyword DOCTYPE must be followed by the name of the root element in the XML document.
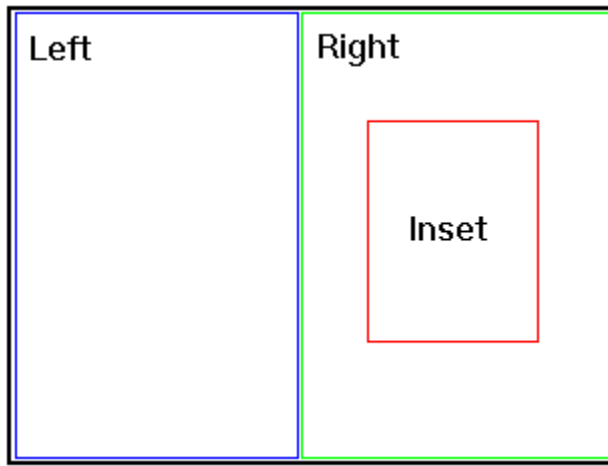
(b) Write a code to handle smil specification.

Code to handle smil specification:

```
<smil >
 <head>
  <paramGroup xml:id="clown">
    <param name="mood" value="upBeat" valuetype="data"/>
    <param name="accessories" value="flowers,dunceCap"/>
  </paramGroup>
 </head>
 <body>

<layout>
    <topLayout width="640px" height="480px" >
        <region id="left" top="0%" left="0%"
             width="50%" height="100%" />
        <region id="right" top="0%" left="50%"
             width="50%" height="100%">
          <region id="inset" top="25%" left="25%"
               width="50%" height="50%" />
```

```
        </region>
      </topLayout>
   </layout>
  </body>
</smil>
```

```
┌──────────────┬──────────────┐
│ Left         │ Right        │
│              │              │
│              │              │
│              │   ┌──────┐   │
│              │   │      │   │
│              │   │ Inset│   │
│              │   │      │   │
│              │   └──────┘   │
│              │              │
└──────────────┴──────────────┘
```

(c) What is HTML and XHTML?

HTML stands for Hyper Text Markup Language. It was proposed by Tim Berners-Lee in 1987. It is the main markup language for creating web pages and other information that can be displayed in a web browser. It creates structured documents by denoting structural semantics for text like headings, lists, links, quotes etc. It allows images and objects to be embedded to create interactive forms. The filename extension is .html/ .htm. HTML is not case sensitive.

XHTML stands for Extensible Hyper Text Markup Language. Its origin was World Wide Web Consortium (W3C). **XHTML** is a family of XML languages which extend or mirror versions of HTML. It does not allow omission of any tags or use of attribute minimization. XHTML requires that there be an end tag to every start tag and all nested tags must be closed in the right order. The filename extension is .xhtml, .xht, .xml, .html, .htm. XHTML is case sensitive.

(d) Draw a graphic using svg2.

A Graphic using SVG2: (https://editor.method.ac/)
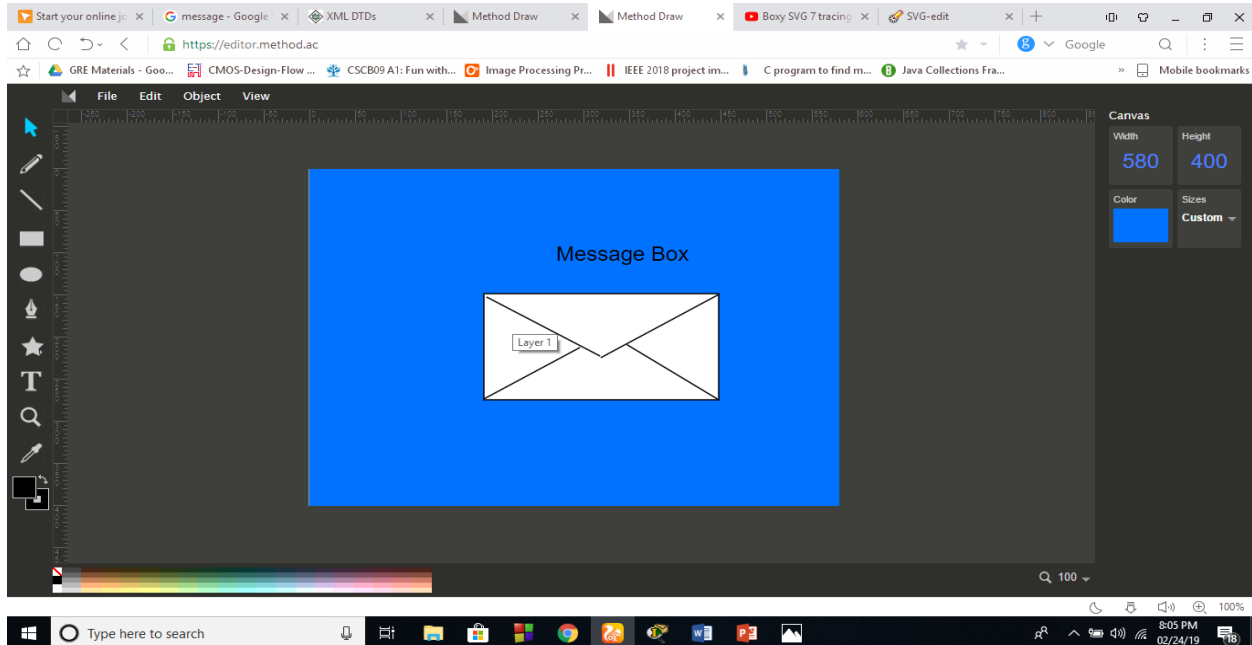
(e) Define Human Visual Dynamics.

Human Visual Dynamics:

The Human Visual Dynamic (HVD) is a system that attempts to select salient areas to reduce cognitive processing efforts. Computational models of visual attention try to predict the most relevant and important areas of videos or images viewed by the human eye. Such models, in turn, can be applied to areas such as computer graphics, video coding, and quality assessment. Although several models have been proposed, only one of them is applicable to high dynamic range (HDR) image content, and no work has been done for HDR videos. Moreover, the main shortcoming of the existing models is that they cannot simulate the characteristics of HVD under the wide luminous range found in HDR content. This paper addresses these issues by presenting a computational approach to model the bottom-up visual saliency for HDR input by combining spatial and temporal visual features. An analysis of eye movement data affirms the effectiveness of the proposed model. Comparisons employing three well-known quantitative metrics show that the proposed model substantially improves predictions of visual attention for HDR content.

(f) Run "Hello World" on IBM Quantum Computer.

IBM Quantum Computer:

```python
from qiskit import QuantumProgram

# Create a QuantumProgram object instance.

qp = QuantumProgram()

# Create a Quantum Register called "qr" with 2 qubits.

qr = qp.create_quantum_register('qr',2)

# Create a Classical Register called "cr" with 2 bits.

cr = qp.create_classical_register('cr',2)

# Create a Quantum Circuit called "qc" involving qr and cr.

qc = qp.create_circuit('HelloWorldCircuit', [qr],[cr])

# Setup connection with real Quantum chip

backend = 'ibmqx5'

#Get you token from https://quantumexperience.ng.bluemix.net/qx/account/advanced

token = 'a7dbfb3cfc1252c4a7555020c32808cff17102a467c595801371f7b7f1f7c3a3355d565469aa4a37564df269f3710f33d7d13ba3c900ca947c1513598b64c5e7'

qp.set_api(token,url='https://quantumexperience.ng.bluemix.net/api')

# Add the H gate in the Qubit 1, putting this qubit in superposition.

qc.h(qr[1])

# Add the CX gate on control qubit 1 and target qubit 0, putting the qubits in a Bell state i.e entanglement

qc.cx(qr[1], qr[0])

# Add a Measure gate to see the state.

qc.measure(qr[0],cr[0])

qc.measure(qr[1],cr[1])

# Compile and execute the Quantum Program in the ibmqx5

results = qp.execute(['HelloWorldCircuit'] ,backend ,timeout=2400)

print(results.get_counts('HelloWorldCircuit'))
```

(g) Write a short note on Font and its history.

Font and Its History:

A font is a set of printable or displayable text character s in a specific style and size. The type design for a set of fonts is the typeface and variations of this design form the *typeface family*. Thus, Helvetica is a typeface family, Helvetica italic is a typeface, and Helvetica italic 10-point is a font. In practice, *font* and *typeface* are often used without much precision, sometimes interchangably.

History:

Type foundries have cast fonts in lead alloys from the 1450s until the present, although wood served as the material for some large fonts called wood type during the 19th century, particularly in the United States. In the 1890s the mechanization of typesetting allowed automated casting of fonts on the fly as lines of type in the size and length needed. This was known as continuous casting, and remained profitable and widespread until its demise in the 1970s. The first machine of this type was the Linotype machine, invented by Ottmar Mergenthaler.