****

1) Create an XML document showing the marksheet of a student of MCA IV sem Gujarat university. It must contain proper elements, attributes, general entities and CDATA sections.

****

****

Students.xml

****

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE studinfo [
  <!ELEMENT studinfo (footer,(student)*)>

  <!ELEMENT footer (#PCDATA)>
  <!ELEMENT student (name,rollno,dept,class,semester,(subject)*)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT rollno (#PCDATA)>
  <!ELEMENT dept (#PCDATA)>
  <!ELEMENT class (#PCDATA)>
  <!ELEMENT semester (#PCDATA)>
  <!ELEMENT subject (subname,marks)>
  <!ELEMENT subname (#PCDATA)>
  <!ELEMENT marks (#PCDATA)>
  <!ATTLIST student id CDATA #REQUIRED>
  <!ATTLIST student color CDATA #REQUIRED>

  <!ENTITY copyright "Copyright:Rcc">
]>
<studinfo>
<![CDATA[Student Matksheet!]]>
        <student id="1" color="red">
                <name>Monik</name>
                <rollno>14</rollno>
                <dept>IT</dept>
                <class>Mca</class>
                <semester>4</semester>
                <subject>
                        <subname>Java</subname>
                        <marks>72</marks>
                </subject>
                <subject>
                        <subname>C++</subname>
                        <marks>78</marks>
                </subject>
        </student>
                <student id="2" color="yellow">
                <name>Manan</name>
                <rollno>50</rollno>
                <dept>IT</dept>
                <class>Mca</class>
                <semester>4</semester>
                <subject>
                        <subname>Java</subname>
```

```
                <marks>50</marks>
            </subject>
            <subject>
                <subname>C++</subname>
                <marks>72</marks>
            </subject>
        </student>
        <footer>&copyright;</footer>
</studinfo>
```

****

2) Create an application that stores name/number pairs for a simple phone book, using an XML format for the data. You can use the XML format illustrated in the following sample phone directory file: Use the DOM API to read the data for XML parsing and display the info in comma separated text file.

```
            <?xml version="1.0"?>
            <phone directory>
                <entry name=’ram’ place=’home’ number=’28901203’/>
                <entry name=’shyam’ place=’mobile’ number=’45559923’/>
            </phone directory>
```

****

****
phone_dir.xml
****

```
<?xml version="1.0"?>
<phone_dir>
<entry name="ram" place="home" number="28901203"/>
<entry name="shyam" place="mobile" number="45559923"/>
</phone_dir>
```

****
phone_dir.java
****

```
public class phone_dir
{
        private String name;
        private String place;
        private int number;
        public phone_dir()
        {
                name=null;
                place=null;
                number=0;
        }
        public phone_dir(String name,String place,int number)
        {
                this.name=name;
```

```
		this.place=place;
		this.number=number;
	}
	public void setname(String name)
	{
		this.name=name;
	}
	public void setplace(String place)
	{
		this.place=place;
	}
	public void setnumber(int number)
	{
		this.number=number;
	}
	public String toString()
	{
		return "Name: "+name+", Place: "+place+", Number: "+number;
	}
}
```

****
que_2.java
****

```
import java.io.*;
import java.util.*;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
public class que_2
{
	public static void main(String args[])throws ParserConfigurationException,IOException,SAXException
	{
		DocumentBuilderFactory dbf=DocumentBuilderFactory.newInstance();
		DocumentBuilder db=dbf.newDocumentBuilder();
		Document d=db.parse("phone_dir.xml");
		List<phone_dir> l=new ArrayList<phone_dir>();
		NodeList nl=d.getDocumentElement().getChildNodes();
		for(int i=0; i<nl.getLength();i++)
		{
			Node n=nl.item(i);
			if(n.getNodeType()==Node.ELEMENT_NODE)
			{
				Element el=(Element)n;
				String name,place;
				int number;
				name=n.getAttributes().getNamedItem("name").getNodeValue();
				place=n.getAttributes().getNamedItem("place").getNodeValue();
```

```
        number=Integer.parseInt(n.getAttributes().getNamedItem("number").getNodeValue());

                                l.add(new phone_dir(name,place,number));
                        }
                }
                FileWriter fw=new FileWriter("output.txt");
                BufferedWriter bw=new BufferedWriter(fw);
                for(phone_dir empl:l)
                {
                        System.out.println(empl.toString());
                        bw.write(empl.toString()+"\n");
                }
                bw.close();
        }
}
```

****
Output.txt
****

```
Name: ram, Place: home, Number: 28901203
Name: shyam, Place: mobile, Number: 45559923
```

****
Output :
****

```
E:\MCA\XML\javac que_2.java

E:\MCA\XML\java que_2
Name: ram, Place: home, Number: 28901203
Name: shyam, Place: mobile, Number: 45559923
```

****

3) Create an application that stores name/number pairs for a simple phone book, using an XML format for the data. Use the DOM API to read the data for XML parsing and display the info in comma separated text file.

****

****
phone_dir.xml
****

```
<?xml version="1.0"?>
<phone_dir>
<entry name="ram" place="home" number="28901203"/>
<entry name="shyam" place="mobile" number="45559923"/>
</phone_dir>
```

****
que_4.java
****

```
import java.io.*;
import java.util.*;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
public class que_4
{
	public static void main(String args[])throws ParserConfigurationException,IOException,SAXException
	{
		DocumentBuilderFactory dbf=DocumentBuilderFactory.newInstance();
		DocumentBuilder db=dbf.newDocumentBuilder();
		Document d=db.parse("pd.xml");
		Hashtable<String,Integer> l=new Hashtable<String,Integer>();
		NodeList nl=d.getDocumentElement().getChildNodes();
		for(int i=0; i<nl.getLength();i++)
		{
			Node n=nl.item(i);
			if(n.getNodeType()==Node.ELEMENT_NODE)
			{
				Element el=(Element)n;
				String name,place;
				int number;
				name=n.getAttributes().getNamedItem("name").getNodeValue();
				place=n.getAttributes().getNamedItem("place").getNodeValue();

	number=Integer.parseInt(n.getAttributes().getNamedItem("number").getNodeValue());

				l.put(name,number);
			}
		}
		FileWriter fw=new FileWriter("output.txt");
		BufferedWriter bw=new BufferedWriter(fw);
		for(Map.Entry empl:l.entrySet())
		{
			System.out.println(empl.getKey()+","+empl.getValue());
			bw.write(empl.getKey()+","+empl.getValue()+"\n");
		}
		bw.close();
	}
}
```

****
Output.txt
****
shyam,45559923
ram,28901203

****
Output :
****

E:\MCA\XML\javac que_4.java

E:\MCA\XML\java que_4

shyam,45559923
ram,28901203

****

4) Below is the XML file which contains information about circles

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="test.css"?>
<!-- It's an xml-stylesheet processing instruction. -->
<!DOCTYPE shapes SYSTEM "shapes.dtd">
<shapes>
      ……
      <squre color="BLUE">
          <length> 20 </length>
      </squre>
      ……
 </shapes>
```

```
<?xml version="1.0"?>
<!DOCTYPE shapes SYSTEM “shapes.dtd">
<shapes>
        <circle color=“BLUE”>
            <x> 20 </x>
            <y> 20 </y>
            <radius> 20 </radius>
        </circle>
</shapes>
```

Extract all the information about **circles** using DOMParser and SAXParser. Your program should throw error if xml file does not exist.

****

****
Shapes.xml
****

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

<shapes>
        <circle color="BLUE">
                <x>20</x>
                <y>20</y>
                <radius>20</radius>
        </circle>
        <square color="BLUE">
                <length>20</length>
        </square>
        <rectangle color="BLUE">
                <length>20</length>
                <breadth>10</breadth>
        </rectangle>
        <circle color="RED">
                <x>30</x>
                <y>30</y>
                <radius>10</radius>
        </circle>
</shapes>
```

****
Shapes.java
****

```
public class shapes
{
```

```
	private String color;
	private int x,y,r;
	private static int c=0;
	public shapes()
	{
		color=null;
		x=0;
		y=0;
		r=0;
	}
	public shapes(String color,int x,int y,int r)
	{
		this.color=color;
		this.x=x;
		this.y=y;
		this.r=r;
	}
	public void setcolor(String color)
	{
		this.color=color;
	}
	public void setx(int x)
	{
		this.x=x;
	}
	public void sety(int y)
	{
		this.y=y;
	}
	public void setr(int r)
	{
		this.r=r;
	}
	public String toString()
	{
		c++;
		return "Circle number-"+c+" color: "+color+", X: "+x+", Y: "+y+" Radius: "+r;
	}
}
```

****

Que_6.java

****

```
import java.io.*;
import java.util.*;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
```

```
public class Que_6
{
	public static void main(String args[])throws ParserConfigurationException,IOException,SAXException
	{
		DocumentBuilderFactory dbf=DocumentBuilderFactory.newInstance();
		DocumentBuilder db=dbf.newDocumentBuilder();
		Document d=db.parse("shapes.xml");
		List<shapes> l=new ArrayList<shapes>();
		NodeList nl=d.getDocumentElement().getChildNodes();
		for(int i=0; i<nl.getLength();i++)
		{
			Node n=nl.item(i);
			if(n.getNodeType()==Node.ELEMENT_NODE)
			{
				Element el=(Element)n;
				System.out.println("Element name: "+el.getTagName());
				if(el.getTagName().equals("circle"))
				{
				String color;
				int x,y,r;
				color=n.getAttributes().getNamedItem("color").getNodeValue();

	x=Integer.parseInt(el.getElementsByTagName("x").item(0).getChildNodes().item(0).getNodeValue());

	y=Integer.parseInt(el.getElementsByTagName("y").item(0).getChildNodes().item(0).getNodeValue());

	r=Integer.parseInt(el.getElementsByTagName("radius").item(0).getChildNodes().item(0).getNodeValue());
					l.add(new shapes(color,x,y,r));
					}
			}
		}
		for(shapes empl:l)
		{
			System.out.println(empl.toString());
		}
	}
}
```

****
Que6_SAX.java
****

```
import java.io.*;
import java.util.*;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;
import org.xml.sax.Attributes;
import org.xml.sax.helpers.DefaultHandler;
import org.xml.sax.SAXException;

public class Que6_SAX extends DefaultHandler
{
```

```
private static List<shapes> emps=new ArrayList<shapes>();
private static shapes emp=null;
boolean x=false,y=false,r=false;

public void startElement(String uri,String localName,String qName,Attributes att)throws SAXException
{
	if(qName.equalsIgnoreCase("circle"))
	{
		emp=new shapes();
		String color=att.getValue("color");
		emp.setcolor(color);
	}
	else if(qName.equalsIgnoreCase("x"))
	{
		x=true;
	}
	else if(qName.equalsIgnoreCase("y"))
	{
		y=true;
	}
	else if(qName.equalsIgnoreCase("radius"))
	{
		r=true;
	}
}
public void endElement(String uri,String localName,String qName)throws SAXException
{
	if(qName.equalsIgnoreCase("circle"))
	{
		emps.add(emp);
	}
}
public void characters(char[] ch,int s,int l)throws SAXException
{
	if(x)
	{
		emp.setx(Integer.parseInt(new String(ch,s,l)));
		x=false;
	}
	else if(y)
	{
		emp.sety(Integer.parseInt(new String(ch,s,l)));
		y=false;
	}
	else if(r)
	{
		emp.setr(Integer.parseInt(new String(ch,s,l)));
		r=false;
	}
}

public static void main(String args[]) throws ParserConfigurationException,SAXException,IOException
{
	Scanner sc=new Scanner(System.in);
```

```
                String xname;
                //System.out.println("Enter XML File name: ");
                xname="shapes.xml";
                SAXParserFactory spf=SAXParserFactory.newInstance();
                SAXParser sp=spf.newSAXParser();
                Que6_SAX s=new Que6_SAX();
                sp.parse(new File(xname),s);
                //sp.parse(xname,s);
                for(shapes e:emps)
                {
                        System.out.println(e.toString());
                }
        }
}
```

****

Output

****

```
E:\MCA\XML\javac Que_6.java

E:\MCA\XML\java Que_6
Element name: circle
Element name: square
Element name: rectangle
Element name: circle
Circle number-1 color: BLUE, X: 20, Y: 20 Radius: 20
Circle number-2 color: RED, X: 30, Y: 30 Radius: 10

E:\MCA\XML\javac Que6_SAX.java

E:\MCA\XML\java Que6_SAX
Circle number-1 color: BLUE, X: 20, Y: 20 Radius: 20
Circle number-2 color: RED, X: 30, Y: 30 Radius: 10

E:\MCA\XML\java Que6_SAX
```