Java Advanced Assignment:

1. **SAX Parser :**

- **MyHandler.java :**

```
package com.accolite.xml.sax;

import java.util.ArrayList;
import java.util.List;
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;
public class MyHandler extends DefaultHandler {
	private List<Employee> empList = null;
	private Employee emp = null;
	private StringBuilder data = null;
	public List<Employee> getEmpList() {
		return empList;
	}
	boolean bfName = false;
	boolean blName = false;
	boolean bAge = false;
	boolean bGender = false;
	boolean blocation = false;
	@Override
	public void startElement(String uri, String localName, String qName, Attributes
attributes) throws SAXException {
		if (qName.equalsIgnoreCase("Employee")) {
			String id = attributes.getValue("id");
			emp = new Employee();
			emp.setId(Integer.parseInt(id));
			if (empList == null)
				empList = new ArrayList<>();
		} else if (qName.equalsIgnoreCase("firstName")) { variables
			bfName = true;
		} else if (qName.equalsIgnoreCase("lastName")) {variables
			blName = true;
		} else if (qName.equalsIgnoreCase("age")) {
			bAge = true;
		} else if (qName.equalsIgnoreCase("gender")) {
			bGender = true;
		} else if (qName.equalsIgnoreCase("location")) {
			blocation = true;
		}
		data = new StringBuilder();
	}
```

```
	@Override
	public void endElement(String uri, String localName, String qName) throws
SAXException {
		if (bAge) {
			emp.setAge(Integer.parseInt(data.toString()));
			bAge = false;
		} else if (bfName) {
			emp.setfirstName(data.toString());
			bfName = false;
		} else if (blName) {
			emp.setlastName(data.toString());
			blName = false;
		} else if (blocation) {
			emp.setLocation(data.toString());
			blocation = false;
		} else if (bGender) {
			emp.setGender(data.toString());
			bGender = false;
		}

		if (qName.equalsIgnoreCase("Employee")) {
			empList.add(emp);
		}
	}
	@Override
	public void characters(char ch[], int start, int length) throws SAXException {
		data.append(new String(ch, start, length));
	}
}
```

- **XMLParserSAX.java :**

```
package com.accolite.xml.sax;
import com.accolite.annotations.*;
import java.io.File;
import java.io.IOException;
import java.util.List;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;
import org.xml.sax.SAXException;
public class XMLParserSAX {
   public static void main(String[] args) {
   SAXParserFactory saxParserFactory = SAXParserFactory.newInstance();
   try {
      SAXParser saxParser = saxParserFactory.newSAXParser();
```

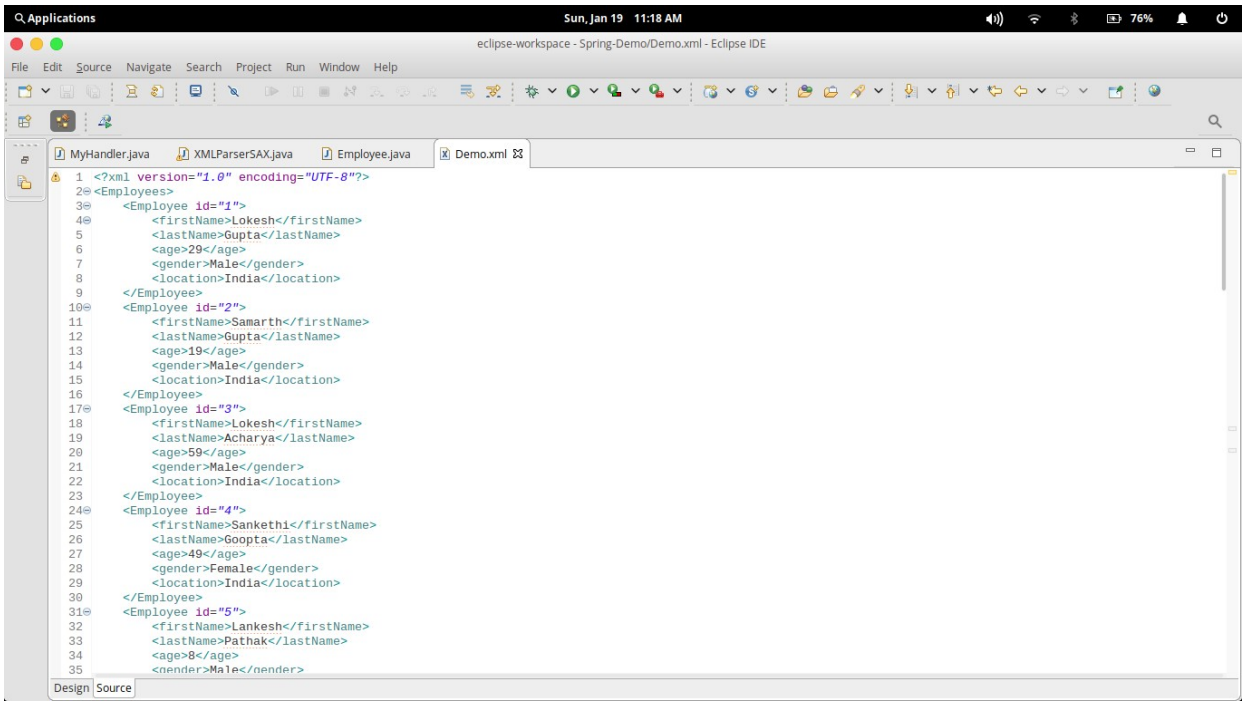
```
        MyHandler handler = new MyHandler();
        saxParser.parse(new File("Demo.xml"), handler);
        List<Employee> empList = handler.getEmpList();
        for(Employee emp : empList)
            System.out.println(emp);
    } catch (ParserConfigurationException | SAXException | IOException e) {
        e.printStackTrace();
    }
    }
}
```

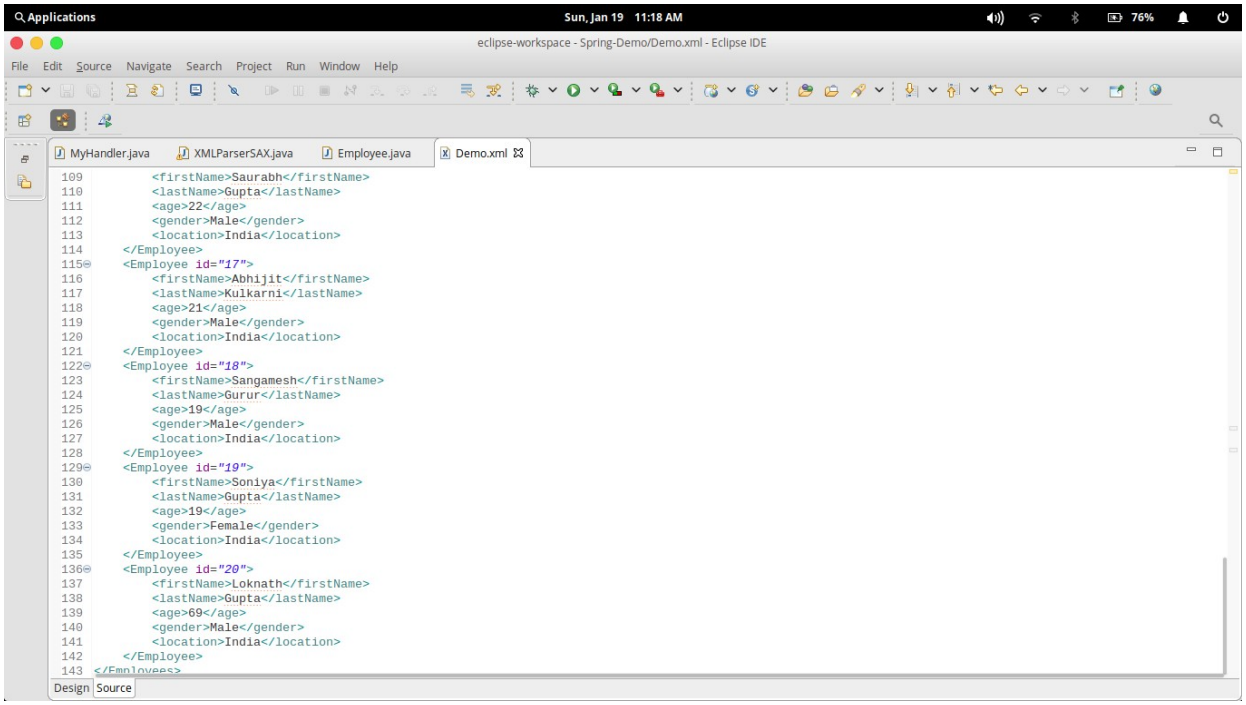
- **Employee.java :**

```
package com.accolite.xml.sax;
public class Employee {
        private int id;
        private String firstName;
        private String lastName;
        private int age;
        private String gender;
        private String location;
        public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }
        public String getfirstName() {
                return firstName;
        }
        public void setlastName(String lastName) {
                this.lastName = lastName;
        }
        public String getlastName() {
                return lastName;
        }
        public void setfirstName(String firstName) {
                this.firstName = firstName;
        }
        public String getGender() {
                return gender;
        }
        public void setGender(String gender) {
                this.gender = gender;
        }
```

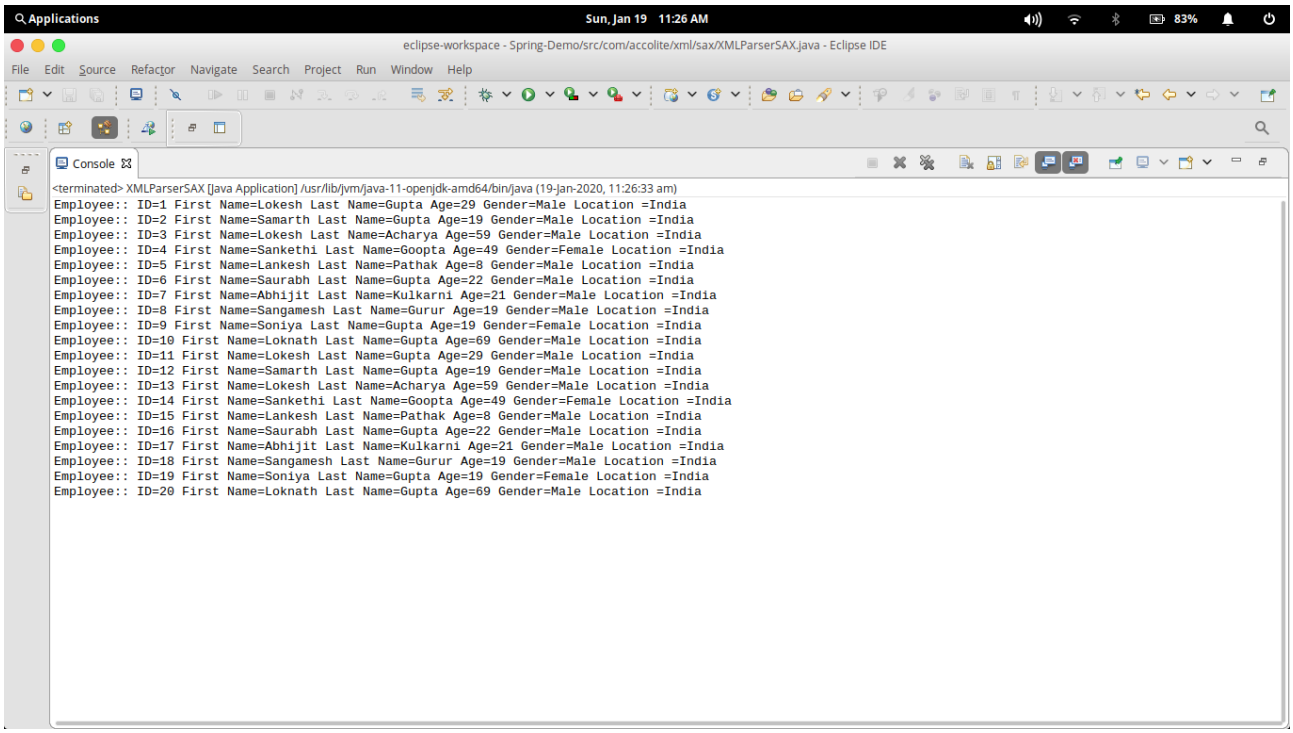
```
        public int getAge() {
                return age;
        }
        public void setAge(int age) {
                this.age = age;
        }
        public String getLocation() {
                return location;
        }
        public void setLocation(String location) {
                this.location = location;
        }
        @Override
        public String toString() {
                return "Employee:: ID=" + this.id + " First Name=" + this.firstName +" Last
Name=" + this.lastName + " Age=" + this.age + " Gender=" + this.gender
                                + " Location =" + this.location;
        }

}
```

- **Input File :**




Applications
Sun, Jan 19 11:18 AM
76%
eclipse-workspace - Spring-Demo/Demo.xml - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help
MyHandler.java
XMLParserSAX.java
Employee.java
Demo.xml
109 <firstName>Saurabh</firstName>
110 <lastName>Gupta</lastName>
111 <age>22</age>
112 <gender>Male</gender>
113 <location>India</location>
114 </Employee>
115 <Employee id="17">
116 <firstName>Abhijit</firstName>
117 <lastName>Kulkarni</lastName>
118 <age>21</age>
119 <gender>Male</gender>
120 <location>India</location>
121 </Employee>
122 <Employee id="18">
123 <firstName>Sangamesh</firstName>
124 <lastName>Gurur</lastName>
125 <age>19</age>
126 <gender>Male</gender>
127 <location>India</location>
128 </Employee>
129 <Employee id="19">
130 <firstName>Soniya</firstName>
131 <lastName>Gupta</lastName>
132 <age>19</age>
133 <gender>Female</gender>
134 <location>India</location>
135 </Employee>
136 <Employee id="20">
137 <firstName>Loknath</firstName>
138 <lastName>Gupta</lastName>
139 <age>69</age>
140 <gender>Male</gender>
141 <location>India</location>
142 </Employee>
143 </Employees>
Design Source

- **Output :**


Applications
Sun, Jan 19 11:26 AM
83%
eclipse-workspace - Spring-Demo/src/com/accolite/xml/sax/XMLParserSAX.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console
<terminated> XMLParserSAX [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (19-Jan-2020, 11:26:33 am)
Employee:: ID=1 First Name=Lokesh Last Name=Gupta Age=29 Gender=Male Location =India
Employee:: ID=2 First Name=Samarth Last Name=Gupta Age=19 Gender=Male Location =India
Employee:: ID=3 First Name=Lokesh Last Name=Acharya Age=59 Gender=Male Location =India
Employee:: ID=4 First Name=Sankethi Last Name=Goopta Age=49 Gender=Female Location =India
Employee:: ID=5 First Name=Lankesh Last Name=Pathak Age=8 Gender=Male Location =India
Employee:: ID=6 First Name=Saurabh Last Name=Gupta Age=22 Gender=Male Location =India
Employee:: ID=7 First Name=Abhijit Last Name=Kulkarni Age=21 Gender=Male Location =India
Employee:: ID=8 First Name=Sangamesh Last Name=Gurur Age=19 Gender=Male Location =India
Employee:: ID=9 First Name=Soniya Last Name=Gupta Age=19 Gender=Female Location =India
Employee:: ID=10 First Name=Loknath Last Name=Gupta Age=69 Gender=Male Location =India
Employee:: ID=11 First Name=Lokesh Last Name=Gupta Age=29 Gender=Male Location =India
Employee:: ID=12 First Name=Samarth Last Name=Gupta Age=19 Gender=Male Location =India
Employee:: ID=13 First Name=Lokesh Last Name=Acharya Age=59 Gender=Male Location =India
Employee:: ID=14 First Name=Sankethi Last Name=Goopta Age=49 Gender=Female Location =India
Employee:: ID=15 First Name=Lankesh Last Name=Pathak Age=8 Gender=Male Location =India
Employee:: ID=16 First Name=Saurabh Last Name=Gupta Age=22 Gender=Male Location =India
Employee:: ID=17 First Name=Abhijit Last Name=Kulkarni Age=21 Gender=Male Location =India
Employee:: ID=18 First Name=Sangamesh Last Name=Gurur Age=19 Gender=Male Location =India
Employee:: ID=19 First Name=Soniya Last Name=Gupta Age=19 Gender=Female Location =India
Employee:: ID=20 First Name=Loknath Last Name=Gupta Age=69 Gender=Male Location =India

2. Create some meaningful FULL Annotation for Method and Class target types.

- **TestCustomAnnotation1.java :**

```
package com.accolite.annotations;
import java.lang.reflect.Method;
public class TestCustomAnnotation1 {
public static void main(String args[])throws Exception{
Hello h = new Hello();
Method m = h.getClass().getMethod("sayHello");
MyAnnotation manno = m.getAnnotation(MyAnnotation.class);
System.out.println("Value is: "+manno.age());
}
}
```

- **MyAnnotation.java :**

```
package com.accolite.annotations;
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
public @interface MyAnnotation {
	int age() default 0;
}
class Hello{
@MyAnnotation(age=10)
public void sayHello(){System.out.println("Hello from annotation");}
}
```

3. Design your own TriConsumer lambda.

```
@FunctionalInterface
interface Triconsumer<T1,T2,T3>{
	void accept(T1 t1 ,T2 t2,T3 t3);
}
public class triconsumerlambda {
  public static void main(String args[]) {
		Triconsumer<Integer, Integer, Integer> t = (a,b,c) -> System.out.println(a+b+c);
		t.accept(15, 10, 5);
  }
}
```

4. Demonstrate Exception handling in Lambda's using wrapper lambdas.

```
import java.util.Scanner;
public class Exceptionlambdas {

	public static void main(String[] args) {
		// Lambda Function
		float numerator = 0, denominator = 0;
		try (Scanner s = new Scanner(System.in)) {
			System.out.println("Enter the numerator :");
			numerator = s.nextFloat();
			System.out.println("Enter the denominator :");
			denominator = s.nextFloat();
		}
		catch(Exception e) {
			System.out.println("Enter proper input. Initialising numerator and
denominator to zero.");
		}
		process(numerator, denominator, divideWrapper((num, den) ->
System.out.println(num/den)));
	}


public static interface DivideLambda{
	public void divide(float numerator, float denominator);
}

private static void process(float numerator, float denominator, DivideLambda divider) {
	divider.divide(numerator, denominator);
}

private static DivideLambda divideWrapper(DivideLambda dividelambda) {
	return (numerator,denominator) -> {
		try {
			if(denominator == 0 )
				throw new ArithmeticException();
			System.out.println(numerator/denominator);
		}
		catch(ArithmeticException e) {
			System.out.println("Can't Divide by Zero.");
		}
	};
}
}
```

5. Create 2 tables in db having some kind of relationship, create a stored procedure which joins the two tables and returns columns in any db and call it using callable statements and map it to a model object.

```
import java.sql.*;

public class databaseJDBC {
   public static void main(String[] args) {
         Connection connection = null;

         try {
           connection = DriverManager.getConnection("jdbc:mysql://localhost/demo?" +
"user=root&password=swarev123");
           Statement statement = connection.createStatement();
           ResultSet resultSet = statement.executeQuery("{call DemoProcedure()}");
           while(resultSet.next())
           System.out.println(resultSet.getInteger(1)+" "+resultSet.getString(2)+"
"+rs.getInteger(3)+" "+rs.getInteger(4)+" "+rs.getString(5));
           connection.close();
         } catch (Exception exception) {
            System.out.println("Operation can not be completed. Please try again.");
         }
   }
}
```

Output :

```
mysql> use demo
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table dept(did int, dname  varchar(10), primary key(did));
Query OK, 0 rows affected (0.05 sec)

mysql> create table emp(id int primary key, name varchar(20),did int, foreign key(did) references dept(did));
Query OK, 0 rows affected (0.05 sec)

mysql> insert into dept values(1,"Development");
ERROR 1406 (22001): Data too long for column 'dname' at row 1
mysql> insert into dept values(1,"Dev");
Query OK, 1 row affected (0.01 sec)

mysql> insert into dept values(2,"Testing");
Query OK, 1 row affected (0.01 sec)

mysql> insert into dept values(3,"Hardware");
Query OK, 1 row affected (0.01 sec)

mysql> insert into emp values(1,Abhijit,1);
ERROR 1054 (42S22): Unknown column 'Abhijit' in 'field list'
mysql> insert into emp values(1,"Abhijit",1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into emp values(2,"Akshat",2);
Query OK, 1 row affected (0.01 sec)

mysql> insert into emp values(3,"Karthik",2);
Query OK, 1 row affected (0.01 sec)

mysql> insert into emp values(4,"Sahil",3);
Query OK, 1 row affected (0.01 sec)
```

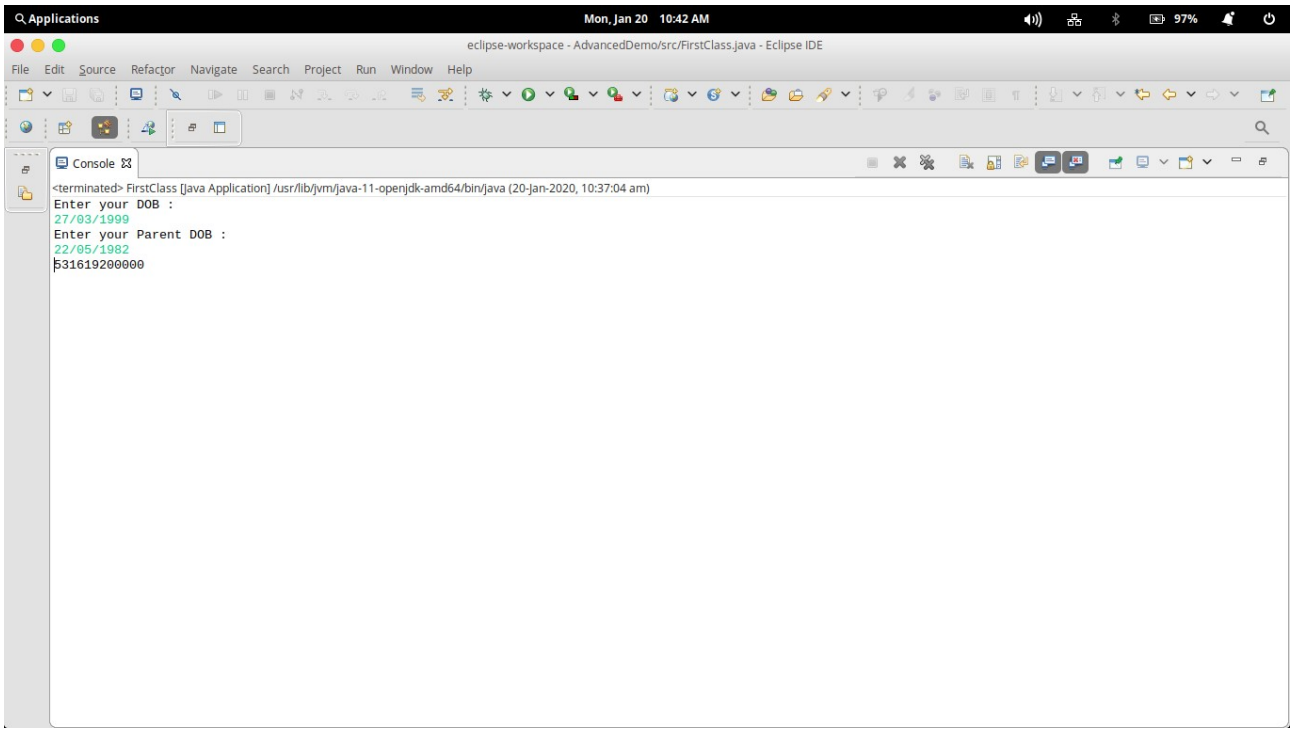
```
mysql> select * from emp;
+----+---------+------+
| id | name    | did  |
+----+---------+------+
|  1 | Abhijit |    1 |
|  2 | Akshat  |    2 |
|  3 | Karthik |    2 |
|  4 | Sahil   |    3 |
+----+---------+------+
4 rows in set (0.00 sec)

mysql> select * from dept;
+-----+----------+
| did | dname    |
+-----+----------+
|   1 | Dev      |
|   2 | Testing  |
|   3 | Hardware |
+-----+----------+
3 rows in set (0.00 sec)

mysql> delimiter //
mysql> create procedure DemoProcedure()
    -> begin
    -> select * from emp join dept on emp.did == dept.did;
    -> end//
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right
syntax to use near '== dept.did;
end' at line 3
mysql> delimiter //
mysql> create procedure DemoProcedure()
    -> begin
    -> select * from emp join dept on emp.did = dept.did;
    -> end//
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> create procedure DemoProcedure()
    -> begin
    -> select * from emp join dept on emp.did == dept.did;
    -> end//
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right
syntax to use near '== dept.did;
end' at line 3
mysql> delimiter //
mysql> create procedure DemoProcedure()
    -> begin
    -> select * from emp join dept on emp.did = dept.did;
    -> end//
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> select call DemoProcedure();
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right
syntax to use near 'call DemoProcedure()' at line 1
mysql> select * from call DemoProcedure();
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right
syntax to use near 'call DemoProcedure()' at line 1
mysql> call DemoProcedure();
+----+---------+------+-----+----------+
| id | name    | did  | did | dname    |
+----+---------+------+-----+----------+
|  1 | Abhijit |    1 |   1 | Dev      |
|  2 | Akshat  |    2 |   2 | Testing  |
|  3 | Karthik |    2 |   2 | Testing  |
|  4 | Sahil   |    3 |   3 | Hardware |
+----+---------+------+-----+----------+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

6. Create a method which takes two strings as inputs (Your dob and your parent/sibling date of birth) and returns the difference in terms of :
1. The number of days of difference between two in nano seconds :

```
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
public class DobNanoseconds {
	public static long getDateDiff(Date date1, Date date2, TimeUnit timeUnit) {
	   long diffInMillies = date2.getTime() - date1.getTime();
	   return timeUnit.convert(diffInMillies,TimeUnit.NANOSECONDS);
	}

	public static void main(String[] args) {
		String myDOB="";
		String parentDOB="";
		Date dateMyDOB = null;
		Date dateParentDOB = null;

		try {
		@SuppressWarnings("resource")
		Scanner s = new Scanner(System.in);
		System.out.println("Enter your DOB :");
		myDOB = s.nextLine();
		System.out.println("Enter your Parent DOB :");
		parentDOB = s.nextLine();
		} catch(Exception e) {
			System.out.println("Can't initiate Scanner.");
		}
		try {
			dateMyDOB = new SimpleDateFormat("dd/MM/yyyy").parse(myDOB);
			dateParentDOB = new
SimpleDateFormat("dd/MM/yyyy").parse(parentDOB);;
		} catch (ParseException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		System.out.println(getDateDiff(dateParentDOB, dateMyDOB,
TimeUnit.NANOSECONDS));
	}

}
```

OUTPUT :



2. Consider your dob in different time zone and then convert your parent/sibling dob in that time zone and find difference in days :

```
import java.text.ParseException;
import java.time.LocalDateTime;
import java.time.ZonedDateTime;
import java.time.temporal.ChronoUnit;
import java.util.Scanner;
public class DobTimezone{

   public static void main(String[] args) throws ParseException
   {
         String MyDOB,ParentDOB;
         try{
         Scanner input = new Scanner(System.in);
         System.out.println("Enter Your DOB in the Format(YYYY-MM-DDTHH:MM:SS) :");
         MyDOB = input.next();
         System.out.println("Enter Your Parent DOB in the Format(YYYY-MM-DDTHH:MM:SS)
:");
         ParentDOB = input.next();
         } Catch(Exception e) {
         System.out.println("Can not read input.");
         }
         LocalDateTime MyDOBTime = LocalDateTime.parse(MyDOB);
         System.out.println("My DateTime: " + MyDOBTime);
```

```
        LocalDateTime ParentDOBTime=LocalDateTime.parse(ParentDOB);
        System.out.println("Parent DateTime: " + ParentDOBTime);
        ZonedDateTime Zone1=ZonedDateTime.parse(MyDOB);
        System.out.println(Zone1);
        ZonedDateTime Zone2=ZonedDateTime.parse(ParentDOB);
        System.out.println(Zone2);
        ChronoUnit CDays=ChronoUnit.DAYS;
        System.out.println(CDays.between(Zone2, Zone1));
    }
}
```

OUTPUT :