1.Name the components that comprise a .java file. List the components in the order that you would expect to see them in a Java program.

1.package declaration(if any):

```
package com.example;
```

2.import statements(if any):

```
import java.util.List;
```

3.class or interface declaration:

```
public class MyClass {
    // Class body
}
```

4.fields(Member variables):

```
private int myField;
```

5.Consturctor:

```
public MyClass() {
    // Constructor body
}
```

6.Methods:

```
public void myMethod() {
    // Method body
}
```

7.Inner classes or interfaces(if any):

```
public class InnerClass {
    // Inner class body
}
```

2.Describe the difference between upper camel case and lower camel case and provide an example of when you would them.

Upper camel case:

```
public class MyClass {
    // Class body
}

public interface MyInterface {
    // Interface body
}

public enum MyEnum {
    VALUE_ONE,
    VALUE_TWO
}
```

Lower camel case:

```
public class MyClass {
    private int myVariable;

    public void myMethod() {
        // Method body
    }

    public static void main(String[] args) {
        MyClass myObject = new MyClass();
        myObject.myMethod();
    }
}
```

3.What syntax is used to import the entire Java utilities package? And if you import an entire package do you also need to import additional classes in the same package separately?

Syntax:

```
import java.util.*;


public class Example {
   public static void main(String[] args) {
      ArrayList<String> list = new ArrayList<>();
      HashMap<String, Integer> map = new HashMap<>();
      Date date = new Date();
   }
}
```

4.Write the syntax for a simple Java object class named Student with the following format: Student Name: Lisa Palombo Student ID: 123456789 Student Status: Active

The student information will be stored in the following variables: fName, lName, stuId, stuStatus

Syntax:

```java
public class Student {
    private String fName;
    private String lName;
    private int stuId;
    private String stuStatus;

    public Student(String fName, String lName, int stuId, String stuStatus) {
        this.fName = fName;
        this.lName = lName;
        this.stuId = stuId;
        this.stuStatus = stuStatus;
    }

    public String getFName() {
        return fName;
    }

    public String getLName() {
        return lName;
    }

    public int getStuId() {
        return stuId;
    }

    public String getStuStatus() {
        return stuStatus;
    }

    public void setFName(String fName) {
        this.fName = fName;
    }

    public void setLName(String lName) {
        this.lName = lName;
    }

    public void setStuId(int stuId) {
        this.stuId = stuId;
    }

    public void setStuStatus(String stuStatus) {
        this.stuStatus = stuStatus;
    }

    public static void main(String[] args) {
        Student student = new Student("Lisa", "Palombo", 123456789, "Active");

        System.out.println("Student Name: " + student.getFName() + " " + student.getLName());
        System.out.println("Student ID: " + student.getStuId());
        System.out.println("Student Status: " + student.getStuStatus());
    }
}
```

Output:

```
Output

java -cp /tmp/dFPIqdCpiz/Student
Student Name: Lisa Palombo
Student ID: 123456789
Student Status: Active

=== Code Execution Successful ===
```

5.Write the code for a Driver Class that will create a Student Object and print the information about the object to the screen.

Syntax:

```java
class Student {
    private String fName;
    private String lName;
    private int stuId;
    private String stuStatus;

    public Student(String fName, String lName, int stuId, String stuStatus) {
        this.fName = fName;
        this.lName = lName;
        this.stuId = stuId;
        this.stuStatus = stuStatus;
    }

    public String getFName() {
        return fName;
    }

    public String getLName() {
        return lName;
    }

    public int getStuId() {
        return stuId;
    }

    public String getStuStatus() {
        return stuStatus;
    }

    public void setFName(String fName) {
        this.fName = fName;
    }

    public void setLName(String lName) {
        this.lName = lName;
    }

    public void setStuId(int stuId) {
        this.stuId = stuId;
    }

    public void setStuStatus(String stuStatus) {
        this.stuStatus = stuStatus;
    }
}
public class StudentDriver {
    public static void main(String[] args) {
        Student student = new Student("America", "Australia", 523101105, "Active");
        System.out.println("Student Name: " + student.getFName() + " " + student.getLName());
        System.out.println("Student ID: " + student.getStuId());
        System.out.println("Student Status: " + student.getStuStatus());
    }
}
```

Output:

```
java -cp /tmp/2IEzGML7xa/StudentDriver
Student Name: America Australia
Student ID: 523101105
Student Status: Active

=== Code Execution Successful ===
```

6. From this lesson, list 10 Java keywords

The 10 java key words from this lesson is:

1.abstract
2.continue
3.default
4.switch
5.package
6.break
7.new
8.public
9.import
10.implements
And etc….

7.The programm done by using the given comments is

Program:

```java
public class Person {
    private String name;
    private int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public static void main(String[] args) {
        Person person = new Person("John Doe", 30);
        System.out.println("Person Name: " + person.getName());
        System.out.println("Person Age: " + person.getAge());
        person.setAge(31);
        System.out.println("Updated Person Age: " + person.getAge());
    }
```

Output:

```
java -cp /tmp/w0GRHdt4oo/Person
Person Name: John Doe
Person Age: 30
Updated Person Age: 31

=== Code Execution Successful ===
```

8. The programm done by using the given comments is

Program:

```java
class Person {
    private String name;
    private int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}
public class ManagingPeople {
    public static void main(String[] args) {
        Person p1 = new Person("Arial", 37);
        Person p2 = new Person("Joseph", 15);
        if (p1.getAge() == p2.getAge()) {
            System.out.println(p1.getName() + " is the same age as " + p2.getName());
        } else {
            System.out.println(p1.getName() + " is NOT the same age as " + p2.getName());
        }
    }
}
```

Output:

```
java -cp /tmp/LOUaOqPHdB/ManagingPeople
Arial is NOT the same age as Joseph

=== Code Execution Successful ===
```