```java
-----------Person.java--------------
package com.ameya.domain;

public class Person implements Comparable<Person>{
    private long id;
    private String firstName;
    private String lastName;
    private int age;
    public Person() {
        super();
        // TODO Auto-generated constructor stub
    }
    public Person(long id, String firstName, String lastName, int age) {
        super();
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.age = age;
    }
    public long getId() {
        return id;
    }
    public void setId(long id) {
        this.id = id;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    @Override
    public String toString() {
        return "Person [id=" + id + ", firstName=" + firstName + ",
        lastName=" +
    lastName + ", age=" + age + "]\n";
    }
    @Override
    public boolean equals(Object obj) {
        return this.id==((Person)obj).getId() ? true : false;
    }
    @Override
    public int hashCode() {
        final long prime=31;
        long result=1;
        result=prime*result+id;
        return (int)result;
    }
    @Override
    public int compareTo(Person o) {
        return ((int)(this.id-o.getId()));
    }
}
---------------FirstNameComparator.java--------------
package com.ameya.utils;

import java.util.Comparator;

import com.ameya.domain.Person;

public class FirstNameComparator implements Comparator<Person>{

    @Override
    public int compare(Person o1, Person o2) {
        return o1.getFirstName().compareTo(o2.getFirstName());
    }

}
-------------AgeComparator.java--------------
package com.ameya.utils;
```

```java

import java.util.Comparator;

import com.ameya.domain.Person;

public class AgeComparator implements Comparator<Person>{

    @Override
    public int compare(Person o1, Person o2) {
        return o1.getAge()-o2.getAge();
    }

}
----------------TestComparators.java--------------
package com.ameya.test;

import java.util.ArrayList;
import java.util.Collections;

import com.ameya.domain.Person;
import com.ameya.utils.AgeComparator;
import com.ameya.utils.FirstNameComparator;

public class TestComparators {

    public static void main(String[] args) {
        ArrayList<Person> list=new ArrayList<Person>();
        list.add(new Person(5,"cccc","cccc",22));
        list.add(new Person(3,"eeee","eeee",27));
        list.add(new Person(4,"aaaa","aaaa",23));
        list.add(new Person(1,"bbbb","bbbb",25));
        list.add(new Person(2,"dddd","dddd",24));

        System.out.println("---------------------------------------------");
        System.out.println("List - No Sorting Criteria");

        System.out.println("---------------------------------------------");
        System.out.println(list);

        System.out.println("---------------------------------------------");
        System.out.println("List - Default Sorting Criteria - on ID");
        System.out.println("---------------------------------------------");
        Collections.sort(list);
        System.out.println(list);
        System.out.println("---------------------------------------------");
        System.out.println("List - Sorting Criteria - on FIRSTNAME");
        System.out.println("---------------------------------------------");
        Collections.sort(list,new FirstNameComparator());
        System.out.println(list);
        System.out.println("---------------------------------------------");
        System.out.println("List - Sorting Criteria - on AGE");
        System.out.println("---------------------------------------------");
        Collections.sort(list,new AgeComparator());
        System.out.println(list);
        System.out.println("---------------------------------------------");
        System.out.println("List - Sorting Criteria - on ID - Descending");
        System.out.println("---------------------------------------------");
        Collections.sort(list,Collections.reverseOrder());
        System.out.println(list);
        System.out.println("List - Sorting Criteria - on FIRSTNAME - Descending");
        System.out.println("---------------------------------------------");
        Collections.sort(list,Collections.reverseOrder(new
```

```java
        FirstNameComparator()));
        System.out.println(list);
        System.out.println("List - Sorting Criteria - on AGE - Descending");
        System.out.println("-------------------------------------------------");
        Collections.sort(list,Collections.reverseOrder(new AgeComparator()));
        System.out.println(list);
    }

}
-----------------USING ANNONYMOUS INNER CLASSES----------
-----------------PersonSortingUtil.java--------------
package com.ameya.utils;

import java.util.Collections;
import java.util.Comparator;
import java.util.List;

import com.ameya.domain.Person;

public class PersonSortingUtil {
    private List<Person> persons;
    public PersonSortingUtil(List<Person> persons) {
        this.persons=persons;
    }
    public void sortOnIdDesc() {
        Collections.sort(persons,Collections.reverseOrder());
    }
    public void sortOnIdAsc() {
        Collections.sort(persons);
    }
    public void sortOnFirstNameAsc() {
        Collections.sort(persons,
                new Comparator<Person>() {
                    @Override
                    public int compare(Person p1,Person p2) {
                        return
                    p1.getFirstName().compareTo(p2.getFirstName());
                }
            });
    }
    public void sortOnAgeAsc() {
        Collections.sort(persons,
                new Comparator<Person>() {
                    @Override
                    public int compare(Person p1,Person p2) {
                        return p1.getAge()-p2.getAge();
                    }
                });
    }
    public void sortOnFirstNameDesc() {
        Collections.sort(persons,Collections.reverseOrder(
                new Comparator<Person>() {
                    @Override
                    public int compare(Person p1,Person p2) {
                        return
                        p1.getFirstName().compareTo(p2.getFirstName());
                    }
                }));
    }
    public void sortOnAgeDesc() {
        Collections.sort(persons,Collections.reverseOrder(
                new Comparator<Person>() {
                    @Override
                    public int compare(Person p1,Person p2) {
                        return p1.getAge()-p2.getAge();
                    }
                }));
    }
    public void printPersonsList() {
        System.out.println("============================");
        System.out.println(persons);
        System.out.println("=============================");
    }
}
----------------TestComparatorsAnnonymous.java--------------
package com.ameya.test;

```

```java
import java.util.ArrayList;

import com.ameya.domain.Person;
import com.ameya.utils.PersonSortingUtil;

public class TestComparatorsAnnonymous {

    public static void main(String[] args) {
        ArrayList<Person> list=new ArrayList<Person>();
        list.add(new Person(5,"cccc","cccc",22));
        list.add(new Person(3,"eeee","eeee",27));
        list.add(new Person(4,"aaaa","aaaa",23));
        list.add(new Person(1,"bbbb","bbbb",25));
        list.add(new Person(2,"dddd","dddd",24));
        PersonSortingUtil util=new PersonSortingUtil(list);
        util.sortOnIdAsc();
        util.printPersonsList();
        util.sortOnIdDesc();
        util.printPersonsList();
        util.sortOnFirstNameAsc();
        util.printPersonsList();
        util.sortOnFirstNameDesc();
        util.printPersonsList();
        util.sortOnAgeAsc();
        util.printPersonsList();
        util.sortOnAgeDesc();
        util.printPersonsList();

    }

}
```