Main:

```
package com.torryharris.mainpack;
import com.torryharris.ipack.item;
import java.util.PriorityQueue;
import java.util.Queue;
public class Main {
    public static void main(String[] args) {
        Queue<item>   pq = new PriorityQueue<item>();
        pq.add(new item(154,"Java"));
        pq.add(new item(45,"HTML"));
        pq.add(new item(4,"C"));
        pq.add(new item(154,"C++"));
        pq.add(new item(60,"Java Script"));
        System.out.println("*************************************");
        for (item i:pq)
        {
            System.out.println(i);
        }
        System.out.println(pq.poll());
        System.out.println("*************************************");
        for (item i:pq)
        {
            System.out.println(i);
        }
        System.out.println(pq.poll());
        System.out.println("*************************************");
        for (item i:pq)
        {
            System.out.println(i);
        }





    }
}
```

```
Item class:

package com.torryharris.ipack;
public class item  implements Comparable<item>{
    private int id;
    private String name;
    public item(int id, String name) {
        this.id = id;
        this.name = name;
    }
    public int getId() {
        return id;
    }
    public String getName() {
        return name;
    }
    @Override
    public String toString() {
        return "Item{" +
                "id=" + id +
                ", name='" + name + '\'' +
                '}';
    }
    // the least id value will be given the most priority
    @Override
    public int compareTo(item i) {
        if(id<i.id)
            return 1;
        else
        if (id==i.id)
            return 0;
        else
            return -1;
    }
}
```

```
Output:
*************************************
Item{id=154, name='Java'}
Item{id=154, name='C++'}
Item{id=4, name='C'}
Item{id=45, name='HTML'}
Item{id=60, name='Java Script'}
Item{id=154, name='Java'}
*************************************
Item{id=154, name='C++'}
Item{id=60, name='Java Script'}
Item{id=4, name='C'}
Item{id=45, name='HTML'}
Item{id=154, name='C++'}
*************************************
Item{id=60, name='Java Script'}
Item{id=45, name='HTML'}
Item{id=4, name='C'}
========================================================================================
```

Main:

```java
package com.torryharris.mainpack;

import com.torryharris.spack.student;

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

public class Main {

    public static void main(String[] args) {
   // write your code here
        ArrayList<student> slist = new ArrayList<student>();
        slist.add(new student(10,"xxx",75));
        slist.add(new student(20,"yyy",65));
        slist.add(new student(30,"zzz",55));
        slist.add(new student(40,"vvv",85));
        slist.add(new student(50,"sss",95));
        System.out.println("printing before sorting");
        for(student s:slist)
            System.out.println(s);
        Collections.sort(slist);
        System.out.println("printing after sorting");
        for(student s:slist)
            System.out.println(s);
    }
}
```

Student class:

```java
package com.torryharris.spack;

public class student implements Comparable<student> {
    private int id;
    private String name;
    private int marks;

    public student(int id, String name, int marks) {
        this.id = id;
        this.name = name;
        this.marks = marks;
    }

    @Override
    public String toString() {
        return "student{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", marks=" + marks +
                '}';
    }

    @Override
    public int compareTo(student student) {
        if (marks > student.marks)
            return 1;
        else
            if (marks == student.marks)
                return 0;
            else
                return -1;

    }
}
```

Output:

```
printing before sorting

student{id=10, name='xxx', marks=75}

student{id=20, name='yyy', marks=65}

student{id=30, name='zzz', marks=55}

student{id=40, name='vvv', marks=85}

student{id=50, name='sss', marks=95}

printing after sorting

student{id=30, name='zzz', marks=55}

student{id=20, name='yyy', marks=65}

student{id=10, name='xxx', marks=75}

student{id=40, name='vvv', marks=85}

student{id=50, name='sss', marks=95}
```

===

Main:

```
package com.torryharris.mainpack;

import com.torryharris.comparatorpack.agecomparator;
import com.torryharris.comparatorpack.namecomparator;
import com.torryharris.ppack.person;

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

public class Main {
    public static void main(String[] args) {
   // write your code here
        ArrayList<person> plist = new ArrayList<person>();
        plist.add(new person(10,"varun",91));
        plist.add(new person(20,"suhas",22));
        plist.add(new person(30,"ajay",60));
        plist.add(new person(40,"abhi",45));
        System.out.println(plist);
        Collections.sort(plist,new agecomparator());
        System.out.println("sorted");
        System.out.println(plist);
        Collections.sort(plist,new namecomparator());
        System.out.println(plist);
    }
}
```

namecomparator class:

```
package com.torryharris.comparatorpack;

import com.torryharris.ppack.person;

import java.util.Comparator;

public class namecomparator implements Comparator<person> {

    @Override
    public int compare(person t1, person t2) {
        return (t1.getName().compareTo(t2.getName()));

    }
}
```

person class:

```
package com.torryharris.ppack;

public class person {
    private int id;
    private String name;
    private int age;

    public person(int id, String name, int age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    @Override
    public String toString() {
        return "person{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", age=" + age +
                '}';
    }
}
```

agecomparator class:

```
package com.torryharris.comparatorpack;

import com.torryharris.ppack.person;

import java.util.Comparator;

public class agecomparator implements Comparator<person> {


    @Override
    public int compare(person p1, person p2) {
       if(p1.getAge()> p2.getAge())
           return 1;
       else
           if(p1.getAge()==p2.getAge())
               return 0;
           else
               return -1;
    }
}
```

output:

```
[person{id=10, name='varun', age=91}, person{id=20, name='suhas', age=22}, person{id=30,
name='ajay', age=60}, person{id=40, name='abhi', age=45}]

sorted

[person{id=20, name='suhas', age=22}, person{id=40, name='abhi', age=45}, person{id=30,
name='ajay', age=60}, person{id=10, name='varun', age=91}]

[person{id=40, name='abhi', age=45}, person{id=30, name='ajay', age=60}, person{id=20,
name='suhas', age=22}, person{id=10, name='varun', age=91}]
```

==

Main:

```
package com.torryharris.mainpack;

import com.torryharris.gpack.product;
import com.torryharris.gpack.test;

public class Main {

    public static void main(String[] args) {
   // write your code here
        test<String,String> tab = new test<String,String>("hello","world");
        System.out.println(tab);
        test<Integer,String> s = new test<Integer,String>(100,"varun");
        System.out.println(s);
        test<Integer, product> s1 = new test<Integer,product>(200,new product(1,"bike"));
        System.out.println(s1);
    }
}
```

product class:

```
package com.torryharris.gpack;

public class product {
    private int id;
    private String name;

    public product(int id, String name) {
        this.id = id;
        this.name = name;
    }

    @Override
    public String toString() {
        return "product{" +
                "id=" + id +
                ", name='" + name + '\'' +
                '}';
    }
}
```

test class:

```
package com.torryharris.gpack;

public class test<T,U> {
    private T obj1;
    private U obj2;

    public test(T obj1, U obj2) {
        this.obj1 = obj1;
```

```java
        this.obj2 = obj2;
    }

    @Override
    public String toString() {
        return "test{" +
                "obj1=" + obj1 +
                ", obj2=" + obj2 +
                '}';
    }
}
```

output:

```
test{obj1=hello, obj2=world}

test{obj1=100, obj2=varun}

test{obj1=200, obj2=product{id=1, name='bike'}}
```

===

Main:

```java
package com.torryharris.mainpack;

public class Main {

    public static void main(String[] args) {
   // write your code here
        Integer[] intarr = {3,5,7,9};
        test.printarray(intarr);

        String[] sarr = {"varun","suhas","ajay"};
        test.printarray(sarr);
    }
}
```

test class:

```java
package com.torryharris.mainpack;

public class test {
    public static <T> void printarray(T[] arr)
    {
        for(T ele:arr)
        {
            System.out.print(ele+"  ");
            System.out.println();
        }
    }
}
```

output:

```
3

5
```

```
7

9

varun

suhas

ajay

================================================================================
```

Main:

```
package com.torryharris.mainpack;

import com.torryharris.tpack.test;

import java.util.ArrayList;

public class Main {

    public static void main(String[] args) {
   // write your code here
        ArrayList<Integer> ilist = new ArrayList<Integer>();
        ilist.add(10);
        ilist.add(10);
        ilist.add(10);
        ilist.add(10);
        ilist.add(10);
        System.out.println(test.sumoflist(ilist));
        //Passing the DOUBLE
        ArrayList<Double> ilist1 = new ArrayList<Double>();
        ilist1.add(10.0);
        ilist1.add(15.0);
        ilist1.add(20.0);
        ilist1.add(25.0);
        ilist1.add(30.0);
        System.out.println(test.sumoflist(ilist1));
        ArrayList<Double> plist = new ArrayList<Double>();
        plist.add(10.0);
        plist.add(10.0);
        plist.add(10.0);
        plist.add(10.0);
        plist.add(10.0);
        System.out.println(test.sumoflist1(plist));
        //Unbounded
        ArrayList<Integer> intlist = new ArrayList<Integer>();
        intlist.add(10);
        intlist.add(15);
        intlist.add(20);
        intlist.add(25);
        intlist.add(30);
        System.out.println(test.sumOfList2(intlist));
    }
}
```

test class:

```java
package com.torryharris.tpack;

import java.util.ArrayList;
import java.util.List;

public class test {
    public static Number sumoflist(List<? extends Number> nlist)
    {
        double s = 0.0;
        for (Number n:nlist)
        {
            s+=n.doubleValue();
        }
        return s;
    }
    public static Number sumoflist1(List<? super Double> llist)
    {
        double s=1.0;
        for(Object n:llist)
        {
            s*=((Number)n).doubleValue();
        }
        return s;
    }
    public static Number sumOfList2(List<?> list)
    {
        double s =0.0;
        for(Object n:list)
        {
            s+= ((Number)n).doubleValue();
        }
        return s;
    }

}
```

output:

```
50.0

100.0

100000.0

100.0
```

==

Pom.xml:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.torryharris.bh84</groupId>
    <artifactId>junitdemo1</artifactId>
    <version>1.0-SNAPSHOT</version>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
        </dependency>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>RELEASE</version>
            <scope>test</scope>
        </dependency>
    </dependencies>



    <properties>
        <maven.compiler.source>8</maven.compiler.source>
        <maven.compiler.target>8</maven.compiler.target>
    </properties>

</project>
```

Evenoddclass:

```
public class evenoddclass {
    public boolean isevennumber(int number) {
        if (number % 2 == 0) {
            return true;
        }
        return false;

    }
}
```

eventest:

```
import org.junit.Test;

import static junit.framework.Assert.assertEquals;
import static org.junit.Assert.assertEquals;

public class eventest {
    @Test
    public void test1()
    {
        evenoddclass ab =new evenoddclass();
        assertEquals(true,ab.isevennumber(12));
    }
    @Test
    public void test2()
    {
        evenoddclass ob =new evenoddclass();
        assertEquals(false,ob.isevennumber(17));
    }
}
output:
```

- [root]
- eventest.test1
- Eventest.test2

===

Test class:

```
import org.junit.Test;

import static org.junit.Assert.*;

public class testclass {
    @Test
    public void test()
    {
        int val1=5;
        int val2=6;

        assertTrue(val1<val2);
        assertFalse(val1>val2);

        String str1="aaa";
        String str2="aaa";

        assertSame(str1,str2);

        String[] earr={"one","two","three"};
        String[] aarr={"one","two","three"};

        assertArrayEquals(earr,aarr);
    }

}
output:
```

- [root]
- testclass.test

===

Message:

```
public class messageutil {
    private String message;

    public messageutil(String message) {
        this.message = message;
    }
    public String printmsg()
    {
        System.out.println(message);
        return (message);
    }
    public String salmsg()
    {
        message="hello"+message;
        System.out.println(message);
        return message;
    }

}
test1:

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class test1 {
    String message="RCB";
    messageutil mob = new messageutil(message);

    @Test
    public void test()
    {
        System.out.println("inside test1 testing printmsg method");
        assertEquals(message,mob.printmsg());
    }

}
test2:

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class test2 {
    String message = "RCB";
    messageutil mob = new messageutil(message);

    @Test
    public void test()
    {
        System.out.println("inside test1 testing printmsg method");
        message="hello"+message;
        assertEquals(message,mob.salmsg());
    }

}
```

testsuit:

```
import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Suite.class)
@Suite.SuiteClasses({
        test1.class,
        test2.class
})
public class testsuit {

}
```

output:

```
inside test1 testing printmsg method

RCB

inside test1 testing printmsg method

helloRCB
```

==

File demo main:

```
package com.torryharris.mainpack;

import java.io.File;
import java.io.IOException;

public class Main {

    public static void main(String[] args) {
   // write your code here
        File file1 = new File("C:\\Users\\varun_srinivas\\Desktop\\thfile1.txt");
        try {
            file1.createNewFile();
            System.out.println("file created");
        } catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println(file1.getAbsolutePath());
        System.out.println(file1.getName());
        System.out.println(file1.getParent());

        File dir = new File("C:\\Users\\varun_srinivas\\Documents");
        File[] lfiles = dir.listFiles();
        for(File f:lfiles)
        {
            if(f.isFile())
                System.out.println(f.getName());
        }

    }
}
```

output:

```
file created
C:\Users\varun_srinivas\Desktop\thfile1.txt
thfile1.txt
C:\Users\varun_srinivas\Desktop
desktop.ini
Java 1.txt
Screenshot (10).png
Screenshot (12).png
Screenshot (13).png
Screenshot (14).png
Screenshot (15).png
Screenshot (16).png
Screenshot (2).png
Screenshot (3).png
Screenshot (4).png
Screenshot (5).png
Screenshot (6).png
Screenshot (7).png
Screenshot (8).png
Screenshot (9).png
Varun D S (5941)-SQL 1.docx
Varun Srinivas_5941_Assignment 01.docx
=================================================================================
```

File demo 2 main:

```
package com.torryharris.mainpack;

import java.io.*;

public class Main {
    public static void main(String[] args) {
   // write your code here
        File file1 = new File("C:\\Users\\varun_srinivas\\Desktop\\thfile1.txt");
        try(FileReader f = new FileReader(file1);
        BufferedReader b = new BufferedReader(f);) {
            String str;
            str = b.readLine();
            while (!str.equals("end........."))
            {
                System.out.println(str);
                str=b.readLine();
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

    }
}
output:

good evening

rcb

varun

========================================================================================
```