****

**Lab 5**

**BCS3323 SOFTWARE TESTING & MAINTENANCE**

**SECTION (01A)**

**Dr ABDULRAHMAN AHMED MOHAMMED AL-SEWARI**

Al Qastholani Bin Karebollah (CB13019)

**CB13019.flt**

@FaultFile

/////////////////////////////////////

// Class information

////////////////////////////////////

classname : DetTrian

methodname : triangle

specifier: public

paramtypes : 3

returntype: String

////////////////////////////////////

// Parameter types

////////////////////////////////////

parameter : partypes[0]=Integer.TYPE

parameter : partypes[1]=Integer.TYPE

parameter : partypes[2]=Integer.TYPE

////////////////////////////////////

// Test Case 0

////////////////////////////////////

arglist : arglist[0]=new Integer(20)

arglist : arglist[1]=new Integer(20)

arglist : arglist[2]=new Integer(20)

////////////////////////////////////

// Test Case 1

////////////////////////////////////

arglist : arglist[0]=new Integer(20)

arglist : arglist[1]=new Integer(10)

arglist : arglist[2]=new Integer(30)

////////////////////////////////////

// Test Case 2

////////////////////////////////////

arglist : arglist[0]=new Integer(20)

arglist : arglist[1]=new Integer(30)

arglist : arglist[2]=new Integer(10)

////////////////////////////////////

// Test Case 3

////////////////////////////////////

arglist : arglist[0]=new Integer(20)

arglist : arglist[1]=new Integer(20)

arglist : arglist[2]=new Integer(-5)

////////////////////////////////////

// Test Case 4

////////////////////////////////////

arglist : arglist[0]=new Integer(0)

arglist : arglist[1]=new Integer(10)

arglist : arglist[2]=new Integer(20)

////////////////////////////////////

// Test Case 5

////////////////////////////////////

arglist : arglist[0]=new Integer(1)

arglist : arglist[1]=new Integer(28)

arglist : arglist[2]=new Integer(30)

////////////////////////////////////

// Test Case 6

////////////////////////////////////

arglist : arglist[0]=new Integer(0)

arglist : arglist[1]=new Integer(000)

////////////////////////////////////

// Test Case 7

////////////////////////////////////

arglist : arglist[0]=new Integer(100)

arglist : arglist[1]=new Integer(50)

arglist : arglist[2]=new Integer(30)

////////////////////////////////////

// Test Case 8

////////////////////////////////////

arglist : arglist[0]=new Integer(70)

arglist : arglist[1]=new Integer(90)

arglist : arglist[2]=new Integer(150)

////////////////////////////////////

// Test Case 9

////////////////////////////////////

arglist : arglist[0]=new Integer(1)

arglist : arglist[1]=new Integer(5)

arglist : arglist[2]=new Integer(0)

////////////////////////////////////

// Test Case 10

////////////////////////////////////

arglist : arglist[0]=new Integer(200)

arglist : arglist[1]=new Integer(320)

arglist : arglist[2]=new Integer(500)

////////////////////////////////////

// Test Case 11

////////////////////////////////////

arglist : arglist[0]=new Integer(1110)

arglist : arglist[1]=new Integer(110)

arglist : arglist[2]=new Integer(1000)

////////////////////////////////////

// Test Case 12

////////////////////////////////////

arglist : arglist[0]=new Integer(10)

arglist : arglist[1]=new Integer(10000)

arglist : arglist[2]=new Integer(1000)

////////////////////////////////////

// Test Case 13

////////////////////////////////////

arglist : arglist[0]=new Integer(00000)

arglist : arglist[1]=new Integer(100)

arglist : arglist[2]=new Integer(1000)

////////////////////////////////////

// Test Case 14

////////////////////////////////////

arglist : arglist[0]=new Integer(-110)

arglist : arglist[1]=new Integer(+1000)

arglist : arglist[2]=new Integer(1000)

////////////////////////////////////

// Test Case 15

////////////////////////////////////

arglist : arglist[0]=new Integer(10\*1)

arglist : arglist[1]=new Integer(1000)

arglist : arglist[2]=new Integer(1000)

////////////////////////////////////

// Test Case 16

////////////////////////////////////

arglist : arglist[0]=new Integer(560)

arglist : arglist[1]=new Integer(-1000)

arglist : arglist[2]=new Integer(1000)

////////////////////////////////////

// Test Case 17

////////////////////////////////////

arglist : arglist[0]=new Integer(90)

arglist : arglist[1]=new Integer(001000)

arglist : arglist[2]=new Integer(1000)

////////////////////////////////////

// Test Case 18

////////////////////////////////////

arglist : arglist[0]=new Integer(1120)

arglist : arglist[1]=new Integer(1000)

arglist : arglist[2]=new Integer(1000)

////////////////////////////////////

// Test Case 19

////////////////////////////////////

arglist : arglist[0]=new Integer(100)

arglist : arglist[1]=new Integer(8000)

arglist : arglist[2]=new Integer(1000)

////////////////////////////////////

// Test Case 20

////////////////////////////////////

arglist : arglist[0]=new Integer(12)

arglist : arglist[1]=new Integer(1000)

arglist : arglist[2]=new Integer(100000)

**Output1.log**

=========================================

Test ID = 0

Class = DetTrian

Method = public triangle

arglist[0]=new Integer(20)

arglist[1]=new Integer(20)

arglist[2]=new Integer(20)

result =Equilateral triangle

=========================================

Test ID = 1

Class = DetTrian

Method = public triangle

arglist[0]=new Integer(20)

arglist[1]=new Integer(10)

arglist[2]=new Integer(30)

result =Any triangle

=========================================

Test ID = 2

Class = DetTrian

Method = public triangle

arglist[0]=new Integer(20)

arglist[1]=new Integer(30)

arglist[2]=new Integer(10)

result =Any triangle

=========================================

Test ID = 3

Class = DetTrian

Method = public triangle

arglist[0]=new Integer(20)

arglist[1]=new Integer(20)

arglist[2]=new Integer(-5)

result =Impossible triangle

=========================================

Test ID = 4

Class = DetTrian

Method = public triangle

arglist[0]=new Integer(0)

arglist[1]=new Integer(10)

arglist[2]=new Integer(20)

result =Impossible triangle

=========================================

Test ID = 5

Class = DetTrian

Method = public triangle

arglist[0]=new Integer(1)

arglist[1]=new Integer(28)

arglist[2]=new Integer(30)

result =Impossible triangle

=========================================

Test ID = 7

Class = DetTrian

Method = public triangle

arglist[1]=new Integer(50)

arglist[2]=new Integer(30)

arglist[0]=new Integer(70)

result =Any triangle

=========================================

Test ID = 8

Class = DetTrian

Method = public triangle

arglist[1]=new Integer(90)

arglist[2]=new Integer(150)

arglist[0]=new Integer(1)

result =Impossible triangle

=========================================

Test ID = 9

Class = DetTrian

Method = public triangle

arglist[1]=new Integer(5)

arglist[2]=new Integer(0)

arglist[0]=new Integer(200)

result =Impossible triangle

=========================================

Test ID = 10

Class = DetTrian

Method = public triangle

arglist[1]=new Integer(320)

arglist[2]=new Integer(500)

arglist[0]=new Integer(1110)

result =Impossible triangle

=========================================

Test ID = 12

Class = DetTrian

Method = public triangle

arglist[1]=new Integer(10000)

arglist[2]=new Integer(1000)

arglist[0]=new Integer(00000)

result =Impossible triangle

=========================================

Test ID = 13

Class = DetTrian

Method = public triangle

arglist[1]=new Integer(100)

arglist[2]=new Integer(1000)

arglist[0]=new Integer(-110)

result =Impossible triangle

=========================================

Test ID = 15

Class = DetTrian

Method = public triangle

arglist[1]=new Integer(1000)

arglist[2]=new Integer(1000)

arglist[0]=new Integer(560)

result =Isoceles triangle

=========================================

Test ID = 17

Class = DetTrian

Method = public triangle

arglist[1]=new Integer(001000)

arglist[2]=new Integer(1000)

arglist[0]=new Integer(1120)

result =Any triangle

=========================================

Test ID = 18

Class = DetTrian

Method = public triangle

arglist[1]=new Integer(1000)

arglist[2]=new Integer(1000)

arglist[0]=new Integer(100)

result =Isoceles triangle

**Detrian.java**

class DetTrian

{

public String triangle (int a, int b, int c)

{

int min,med, max;

String str;

if (a>b)

{

max=a;

min = b;

}

else

{

max = b;

min = a;

}

if (c>max)

max = c;

else if (c<max)

min = c;

med = a+b+c-min-max;

if (max>min+med){

str= "Impossible triangle";

System.out.println("result =" +str);

}

else if (max==min){

str= "Equilateral triangle";

System.out.println("result =" +str);

}

else if (max==med||med==min){

str= "Isoceles triangle";

System.out.println("result =" +str);

}

else if (max\*max==min\*min + med\*med){

str="Rightangled triangle";

System.out.println("result =" +str);

}

else{

str="Any triangle";

System.out.println("result =" +str);

}

return str;

}

}

**JTConstant.java**

/\*\*

\* Title: JTstConstant

\* Description: An Automated Java Unit Testing Tool

\* Copyright: Copyright (c) 2005

\* Company: USM

\* @author: Dr Kamal Zuhairi Zamli, Dr Nor Ashidi Mat Isa, Siti Norbaya Azizan

\* @version: 1.0

\*/

public interface JTstConstant

{

// stub filename for code generation

public static String TEMP\_DIRECTORY = "C:\\CB13019\\SampleExecution\\";

public static String LOG\_DIRECTORY = "C:\\CB13019\\SampleExecution\\";

public static String FAULT\_DIRECTORY ="C:\\CB13019\\SampleExecution\\";

public static String CLASS\_DIRECTORY ="C:\\CB13019\\SampleExecution\\";

public static String INSPECTOR\_FILE = TEMP\_DIRECTORY+"inspect.dat";

public static String JAVA\_EXT =".java";

public static String CLASS\_EXT =".class";

public static String LOG =".log";

public static String OUT =".out";

// .class file cannot have directory info

public static String STUB ="stub";

public static String LOGFILE = LOG\_DIRECTORY+"output";

public static String COMPILE\_RESULT = TEMP\_DIRECTORY+"compile";

public static String HEADER\_FILE = TEMP\_DIRECTORY+"header";

public static String BODY\_FILE = TEMP\_DIRECTORY+"body";

public static String KEY\_FILE = TEMP\_DIRECTORY+"key";

// maximum allowable array length

public static final int MAX = 1500;

}

**Output Command Prompt**

