HONG KONG INSTITUTE OF VOCATIONAL EDUCATION

Laboratory 7: Software Testing and Testing Techniques

Module Intended Learning Outcome:

On completion of the module, students are expected to be able to:

- Apply software testing techniques in various software development stages.

TASK:

1. A Call Graph in testing shows the relationship between calling and called methods.

(a) Draw the Call Graph for the following program.

```
class StudentFactory {

      public static Student createStudent( String kind ) {
          if ( kind.equals("FT") )
              return new FullTimeStudent();
          else
              if ( kind.equals("PT") ) return new PartTimeStudent();
          else
              return null;
       }
 }

abstract class Student { public abstract void whoAmI(); }

class FullTimeStudent extends Student {

  public void whoAmI() {
     System.out.println("I am a full-time student!");
```

```
   }

}

class PartTimeStudent extends Student {

   public void whoAmI() {
      System.out.println("I am a part-time student!"); }
   }

public class Test {
   public static void main( String[] args ) {

      Student s = StudentFactory.createStudent(args[0]);
      s.whoAmI();
   }
}
```

(b) Explain how the *Dynamic Binding* in the Polymorphism makes it difficult in determining the exact method called.

(c) List all the possible scenarios when the test program Test.java runs.

2. Given the following JAVA coding.

```
int proc(int a, int b, int x)
{
  if ((a>1) && (b==0))
  {
    x = x/a;
  }
  else if ((a==2)||(x>1))
  {
    x = x+1;
```

```
    }
    return x;
}
```

(a) Draw the flow graph of the given coding;

(b) Find paths to cover ***all statements*** with possible inputs;

(c) Find paths to cover ***branch coverage condition*** with input values and their corresponding paths.

(d) Do you agree that all statements covered imply branch coverage condition? Justify your answer with your answers to part (b) and (c)?

3. Given the following JAVA program coding for the method ***findMethod*** in a Game program:

```
public int findMethod (int a) {

S1  int x = a;
    int y = 25;

    while (x != y)  {        D1

       if (x > y)            D2

S2        x = x - y;
       else

S3        y = x;

    }

S4  return x;

}
```

Note: S1 through S4 are statement nodes and D1 through D2 are decision nodes in the program.

(a) Draw the Data Dependency Graph for the given JAVA program.

(b) Draw the Control Flow Graph for the given JAVA program.

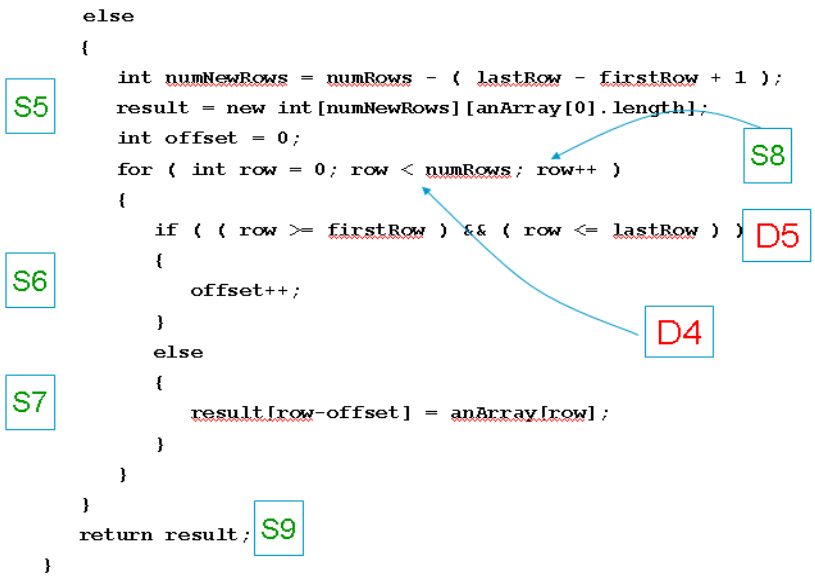
(c) Identify **THREE** execution paths of the Control Flow Graph you answered in (b).

(d) Provide the following details to test the sub-path D2-S3 identified in the Control Flow Graph you answered in (b)

- Extend sub-path D2-S3 to a complete path;
- Find the set of data conditions required to complete the full path;
- Prepare the Equivalence Classes from the data conditions;
- Select an input data from the Equivalence Classes and give the Predict Output.

4. Use the following program to explain why the unit of test coverage can be generated by the statements coverage [S] and decisions coverage [D].

```
public int[][] deleteRows( int[][] anArray,
                           int firstRow, int lastRow )
{
S1  int[][] result = null;
    int numRows = anArray.length;
    if ( ( firstRow >= numRows ) || ( firstRow < 0 ) )   D1
    {
S2      System.out.println( "Bad first row." );
    }
    else if ( ( lastRow >= numRows ) || ( lastRow < 0 ) )   D2
    {
S3      System.out.println( "Bad last row." );
    }
    else if ( lastRow < firstRow )   D3
    {
S4      System.out.println( "Not a valid range." );
    }
```



You are required to complete the following tasks:

(a) Draw the Flow Graph and identify the sub-paths and paths.

(b) Choose the Data Condition and identify its equivalence classes for a path of S1 – D1 – D2 – S3 – S9.

(c) Decide by yourself the Input Data and its Predict Output for the path specified in (b).