White Box Testing :

# Function : Estimate how much portfolio is worth in total

String line // get the data from the txt data file externally

BufferedReader br ;

StringBuilder text ;

**1- Control structure**

Statement coverage :

if br.readLine() != null

{

line = br.readLine();

append data to text ;

print out text the screen ;

}

**2- Basis path** :

Get the live data from the internet

Render it the line variable

Display the line to the interface

**3 - Data Flow diagrams** :

<start the function>

Initialize

**define** : List<Int> listOnAmount

HashMap<String, Double> setOfStockList

List<Double> listOnResult

Object previousData.

Calculate worth result of particular shares

No

Yes

getLiveIndex() and

getLiveValue()

**define** : String temp\_index = getLiveIndex();

double temp\_value = getLiveValue();

**uses** : setOfStockList.put(temp\_index,temp\_value)

**kill** : temp\_index , temp\_value

No

**use :**

double temp = transferToPounds(setOfStockList.get(“BP”))

remove(“BP”)

put(“BP”,temp)

**kill :** temp

if it’s in Pounds

Yes

<end the function>

Print total worth value

No

Calculate the total worth of portfolio

Yes

setOfStockList is not null

**define :**

double temp\_result ;

**use :**

temp\_result = listOnAmount.get(0) \* setOfStockList.get(“BP”);

listOnResult.add(temp\_result);

**kill :** temp\_result

**define :**

double total = 0 ;

TextView total\_output;

**use :**

total += listOfResult.get(i)

append total to total\_output

**kill :** total\_output

Print whole set of stocks in separate lines

Keep track data for future reference

**use :** set Data to previousData object

<end the function>

Basis path - minimum number of basis test cases : 4 tests.

**4 - Data flow anomalies**

- *Write write anomaly* : temp\_index and temp\_value are overwritten when the condition of getLiveIndex() and getLiveValue() occur.

- *Uninitialised variables anomaly :* Add the temp\_index and temp\_value before it’s initialized.

- *Uses variables twice when they were killed :* put dead temp\_index and dead temp\_value tosetOfStockList

Function : How much set of shares that worth in total

List<int> listOnAmount ;

double result ;

List<String> listOnIndex ;

List<Double> listOnResult ;

HashMap <String, double> setOfStockList

Object previousFridayData ; // capture all the things on the last friday

**1- Control structure**

Statement coverage : In case of amount on stocks X is greater than 0, then

calculate the result

if(ammount > 0 && setOfStockList.containsKey(indexList.get(0)) &&

setOfShareList.get(indexList.get(0))) > 0 )

THEN

resultOnX = amountOnX\* setOfStockList.get(indexList.get(0))) ;

Statement : Calculate the total worth of the whole portfolio

double total = resultOnX + resultOnY + resultOnA + resultOnB ;

**2- Basis path** :

Calculate all the result of particular set of stocks

Display results of all set of stocks in separate lines

Calculate total worth of the portfolio

Display total worth of portfolio

**3 - Data Flow diagrams :**

**define :** text file.txt

String line ;

BufferReader br ;

StringBuilder text;

TextView output ;

**use :** line = br.readLine()

Display the TextView output

No

Yes

**use :** append to the text

add to output

while line is not null

Download data from the internet as the text file

<start the function>

<end the function>

Basis path - minimum number of test cases : 2 test cases