Seminar Objectives

- Generating test cases based on white box testing.

Theoretical aspects

- Generating test cases based on white box testing.
- Control Flow Graph
- Coverage criteria: statements, conditions/decisions, paths, loops
- References: [Myers]–chapter 4; [Naik]–chapter 4; [Young]–chapter 12; [Patton] –chapter 6,7

[Myers] Glenford J. Myers, *The Art of Software Testing,* John Wiley & Sons, Inc., 2004
[Naik] K. Naik, P. Tripathy, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008
[Young] M. Pezzand, M. Young, *Software Testing and Analysis: Process, Principles and Techniques*, John Wiley & Sons, 2008
[Patton] R. Patton, Software Testing, Sams Publishing, 2005

Test cases based on source code (CFG, coverage)

Assignment

Based on White-Box Testing develop test cases for the following problems:

//ValueException class

```
public class ValueException extends Exception {
	public ValueException(String msg){
		super(msg);
	}
	public ValueException(Exception ex){
		super(ex);
	}
}
```

1) Verify if a number is prime.

```
public class VerifyIsPrime {
	public VerifyIsPrime(){
		System.out.println("Verify is prime...");
	}
	public boolean isPrime(int n) throws ValueException{
		boolean b = true;
		if(n<0){
			throw new ValueException("data not valid");
		}
		if(n<2){
			b=false;
		}
		else{
			int i=2;
			while (b && (i<= (n/2))){
				if ((n % i) == 0){
					b=false;
				}
				else{
```

```
                                        b=true;
                                }
                                i++;
                        }
                }
                return b;
        }
}
```

2) Compute the maximal sequence of prime numbers from an array of natural numbers. An array X with n components is given.

```
public class LongestPrimeSequence {

        private ArrayList l;
        private int start, length;


         public LongestPrimeSequence(){
              System.out.println("Longest sequence vida ...");
        }
        public boolean isPrime(int n) throws ValueException{ …}

        public void SolveLongestSequence() throws ValueException{
              int posI=-1, lengthI=0, i=0;
              int posF=-1, lengthF=0;
              while(i<this.l.size()){
                    if(isPrime((int)this.l.get(i))==true){
                          if(posI==-1){
                                posI=i;
                                lengthI=1;
                          }
                          else
                                lengthI++;
                    }
                    else{
                          if(lengthI>lengthF){
                                lengthF=lengthI;
                                posF = posI;
                          }
                          posI=-1; lengthI=0;
                    }
                    i++;
              }
              if(lengthI>lengthF){
                    lengthF=lengthI;
                    posF = posI;
              }
        this.start =posF;
        this.length=lengthF;
    }
}
```