BINUS University

Academic Career: ***Undergraduate / ~~Master~~ / ~~Doctoral~~ *)***			**Class Program:** ***~~International~~/Regular/~~Smart Program~~/~~Global Class~~*)***		
☐ **Mid Exam** ☑ **Final Exam** ☐ **Short Term Exam** ☐ **Others Exam : _____**			**Term : ~~Odd~~/Even/~~Short~~ *)**		
☑ **Kemanggisan** ☑ **Alam Sutera** ☑ **Bekasi** ☐ **Senayan** ☐ **Bandung** ☐ **Malang**			**Academic Year :** **2020 / 2021**		
Faculty / Dept.	:	School of Computer Science	Deadline	Day / Date :	Monday / Jul 26th, 2021
				Time :	13:00
Code - Course	:	COMP6640 - Software Engineering	Class	:	All Classes
Lecturer	:	Team	Exam Type	:	Online
*) *Strikethrough the unnecessary items*					
The penalty for CHEATING is DROP OUT!!!					

Learning Outcomes:

LO 2: Explain the software engineering practices and business environment
LO 3: Demonstrate the quality assurances and the potential showcase business project
LO 4: Analyze the software project management and the proposed potential business project
LO 5: Design physical / visual representation and constructive breakthrough of business ideas

I. Essay (40%)

The table given below lists *6 metrics* for each of the *20 methods* of a Java class named 'Directory'.

ClassName	*MethodName*	*Visibility*	*ByteSize*	*#Instructions*	*CyclomaticComplexity*	*#Parameters*	*#LocalVariables*
jdf/tools/io/Directory	<init>	public	48	27	2	1	3
jdf/tools/io/Directory	<init>	public	27	15	1	1	2
jdf/tools/io/Directory	canRead	public	8	4	1	0	1
jdf/tools/io/Directory	canWrite	public	8	4	1	0	1
jdf/tools/io/Directory	exists	public	8	4	1	0	1
jdf/tools/io/Directory	filter	public	99	51	1	1	3
jdf/tools/io/Directory	getAbsolutePath	public	8	4	1	0	1
jdf/tools/io/Directory	getFilter	private	14	10	1	1	4
jdf/tools/io/Directory	getName	public	5	3	1	0	1
jdf/tools/io/Directory	list	public	8	4	1	0	1
jdf/tools/io/Directory	list	public	9	5	1	1	2
jdf/tools/io/Directory	listFiles	public	8	4	1	0	1
jdf/tools/io/Directory	listFiles	public	9	5	1	1	2
jdf/tools/io/Directory	listFiles	public	9	5	1	1	2
jdf/tools/io/Directory	listFiles	public	10	6	1	1	2
jdf/tools/io/Directory	listFiles	public	144	74	8	1	9
jdf/tools/io/Directory	listFiles	public	82	41	4	2	7
jdf/tools/io/Directory	listFiles	public	11	7	1	2	3
jdf/tools/io/Directory	mkdir	public	8	4	1	0	1
jdf/tools/io/Directory	printFiles	private	28	16	3	1	2

The meaning of the 6 metrics is as follows:

- *Visibility*: the visibility modifier of the method – one of private, protected, or public
- *#Instructions*: the number of Bytecode instructions
- *Bytesize*: the total number of Bytes used by all instructions of this method
- *CyclomaticComplexity*: the cyclomatic complexity (number of decision points + 1)

Verified by,

[Kenny Jingga] (D6426) and sent to Program on Jun 23, 2021

- *#Parameters*: the number of formal parameters/arguments
- *#LocalVariables*: the number of local variables declared in the method

As the Java programming language supports *method overloading*, there are 2 methods named 'list' and 7 methods named 'listFiles' that are distinct and all have a different method parameters/arguments.

Answer the following questions and refer back to the table to illustrate/justify your answers:

1. **[LO 2, 6 points]** List the type of scale (Nominal, Ordinal, Interval, Ratio, or Absolute) for each of the 6 metrics.

2. **[LO 2, 4 points]** Do the metrics presented in the table support the statement "*most methods are small*"? Justify your answer.

3. **[LO 2, 5 points]** Comment on how the metrics presented in the table support the statement "*Software is not normal*".

4. **[LO 2 & LO 3, 10 points]** Provide suitable *summary measures* (for example, average/mean, median, standard deviation etc.) for
   a. *#Instructions*,
   b. *CyclomaticComplexity*, and
   c. #LocalVariables

   that provide an *accurate, statistically relevant* summary of the distribution of the corresponding metric.
   Please justify the choice of summary measures and list the corresponding values (e.g., median of *#Parameters* = 1).

5. **[LO 2 & LO 4, 15 points]** You have the time to *thoroughly test* 3 of the 20 methods of the class 'Directory', but no time to test the other 17 methods.
   a. What is the purpose of *software testing*? What is to be uncovered during software testing?
   b. Describe the process how you will choose the 3 methods to test and *justify* why this is a good process
   c. List the 3 methods you will test as the result of applying your process

II. Case (60%)

A freight forwarding company will develop a tracking system application that can provide convenience for consumers in sending goods to various locations. Some of the conveniences of this application are Consumers can send goods with a home pickup system so that consumers just make an order for delivery of goods, then the goods will be picked up and delivered to the destination location. Consumers can check the position of the goods that are on the way with their smartphone as well as the ease of making payments using various online payment systems. In addition, the application can also be used by companies to track vehicles used for shipping goods with real time information on the position, condition of the vehicle, driver's condition and information on the goods being transported.
Based on this case, answer the following questions:

1. **[LO 3, 20 points]** To ensure the application quality for meeting the user needs, identify application metrics that need to be measured and how to measure them to ensure that application functionality meets user needs.

Verified by,

[Kenny Jingga] (D6426) and sent to Program on Jun 23, 2021

2. **[LO 4, 20 points]** Using Function Point analysis, calculate the function points for the completion of the application development project and identify the activities carried out for the development of this application in the form of WBS (work breakdown structure).

3. **[LO 5, 20 points]** If you are asked to lead the development of this application which will be developed in a distributed manner to team members who are in several cities, make the right configuration management software and application architecture model.

-- Good Luck--

Verified by,

[Kenny Jingga] (D6426) and sent to Program on Jun 23, 2021