

Program Content

Semester	I		
Course Code:	IT1406		
Course Name:	Introduction to Programming		
Credit Value:	04		
Core/Optional	Core		
Hourly Breakdown	Theory	Practical	Independent Learning
	45	30	125
Course Aim/Intended Learning Outcomes:			
On completion of this course, students will be able to design and develop programs for specified tasks using Java as an Object Oriented Programming Language.			
Course Content: (Main Topics, Sub topics)			
Topic		Theory (Hrs)	Practical (Hrs.)
1. History and Evolution of Java		1	-
2. Interacting with Java Programming environment		1	1
3. Fundamentals of Java Programming		9	7
4. Computer program design		2	1
5. Object Orientation		8	4
6. Packages and interfaces		2	1
7. Exception handling		3	2
8. Enumerations, Auto boxing and annotations (Meta data)		2	1
9. Understanding generics		1	1
10. Overview of some classes of the Java library		16	12
	Total	45	30
1. History and Evolution of Java (1 hr.)			
Sub Topics			
1.1. Java's Lineage [Ref 1: Pg.(1 – 3)]			
1.2. The birth of Modern Programming : C and C++ [Ref 1: Pg.(4 – 5)]			
1.3. The creation of Java and Microsoft's C# [Ref 1: Pg (6 – 8)]			
1.4. Java and Internet [Ref 1: Pg.(8 – 9)]			
1.4.1. Java Applets			
1.4.2. Security			
1.4.3. Portability			
1.5. The Bytecode [Ref 1: Pg.(9)]			
1.6. Java on the server Side [Ref 1: Pg.(10)]			

- 1.7. Features of Java [Ref 1: Pg.(10 – 13)]
- 1.8. Evolution of Java and Java SE 8 [Ref 1: Pg.(13 – 15)]
- 1.9. Understanding of the Java innovations [Ref 1: Pg.(16)]

2. Interacting with Java Programming environment (1 hr)

Sub Topics

- 2.1. Installing and setting the Java environment in one's computer
- 2.2. First few programs [Ref 1: Pg.(23 – 27), (32 – 33)]
 - 2.2.1. Writing and saving the Java source file
 - 2.2.2. How to compile and execute a Java program
 - 2.2.3. Structure of a Java program
 - 2.2.4. Using blocks of code and issues of lexical in Java at a glance
 - 2.2.4.1. Block of code
 - 2.2.4.2. Whitespace
 - 2.2.4.3. Identifiers
 - 2.2.4.4. Literals
 - 2.2.4.5. Comments
 - 2.2.4.6. Separators
 - 2.2.5. Java keywords and introduction to Java class libraries [Ref 1: Pg.(33 – 34)]

3. Fundamentals of Java Programming (09 hrs)

Sub Topics

- 3.1. Data types and variable in Java
 - 3.1.1. Data types
 - 3.1.1.1. Java is a strongly typed language [Ref 1: Pg.(35)]
 - 3.1.1.2. The primitive types [Ref 1: Pg.(35 – 41)]
 - 3.1.1.3. A closer look at literals [Ref 1: Pg.(41 – 43)]
 - 3.1.1.3.1. Integer literals
 - 3.1.1.3.2. Floating point literals
 - 3.1.1.3.3. Boolean literals
 - 3.1.1.3.4. Character literals and Escape sequences
 - 3.1.1.3.5. String literals
 - 3.1.2. Variables [Ref 1: Pg.(44 – 51)]
 - 3.1.2.1. Variable declarations
 - 3.1.2.2. Conversions in naming variables
 - 3.1.2.3. Dynamic initialization
 - 3.1.2.4. The scope and life time of variables
 - 3.1.2.5. Type conversion and casting
 - 3.1.2.6. Automatic type promotion in expressions
 - 3.1.2.7. The type promotion rules
- 3.2. Operators [Ref 1: Pg.(61 – 80)]
 - 3.2.1. Arithmetic operators
 - 3.2.2. Increment and decrement
 - 3.2.3. Relational operators
 - 3.2.4. Boolean logical operators
 - 3.2.5. The bitwise operators

	3.2.6. The assignment operator
	3.2.7. The? : operator
	3.2.8. Operator precedence
	3.2.9. Usage of parenthesis
3.3.	Control statements [Ref 1: Pg.(81 – 108)]
3.3.1.	Selection statements
3.3.1.1.	If in different forms
3.3.1.2.	Switch in different forms
3.3.2.	Iteration statements
3.3.2.1.	For and for-each(enhanced for) in different forms
3.3.2.2.	Do while in different forms
3.3.2.3.	While in different forms
3.3.3.	Jump statements
3.3.4.1.	Break
3.3.4.2.	Continue
3.3.4.3.	Return
3.4.	Arrays [Ref 1: Pg.(51 – 58)]
3.4.1.	One dimensional
3.4.2.	Multi-dimensional
3.4.3.	Irregular Multi-dimensional (Ragged arrays)
3.4.4.	Using command line arguments and variable length arguments [Ref 1: Pg.(154 – 159)]
4.	Computer program design (2 hrs)
	Sub Topics
4.1.	Steps in program development
4.2.	Program design methodologies
4.2.1.	Procedure-driven
4.2.2.	Event-driven
4.2.3.	Data-driven
4.3.	Procedural versus Object oriented programming
4.4.	An introduction to algorithms and pseudocode
4.5.	Program data
4.6.	Structure theorem
4.7.	First steps in modularisation
4.8.	Communication between modules, cohesion and coupling
4.9.	Flow charts
5.	Object Orientation (8 hrs)
	Sub Topics
5.1.	Introducing classes and objects [Ref 1: Pg.(109 – 126)]
5.2.	Features of Object orientation [Ref 1: Pg.(17 – 22), (161 – 186), (158 – 159)]
5.3.	A closer look at methods and passing data to methods [Ref 1: Pg. (129 – 139)]
5.4.	Recursions [Ref 1: Pg. (139)]
5.5.	Introducing access control [Ref 1: Pg. (141-144)]
5.6.	Understanding static and final key words [Ref 1: Pg.(145 – 146)]

5.7. Nested and inner classes [Ref 1: Pg.(149 – 151)]

6. Packages and interfaces (2 hrs)

Sub Topics

- 6.1. Packages [Ref 1: Pg.(187 – 196)]**
 - 6.1.1. Introduction to package**
 - 6.1.2. Finding packages and CLASSPATH**
 - 6.1.3. Access protection**
 - 6.1.4. Importing a package**
- 6.2. Interfaces [Ref 1: Pg. (196 – 212)]**
 - 6.2.1. Defining an interface**
 - 6.2.2. Implementing interfaces**
 - 6.2.3. Nested interfaces**
 - 6.2.4. Applying interfaces**
 - 6.2.5. Variables in interfaces**
 - 6.2.6. Interfaces can be extended**
 - 6.2.7. Default interface methods**
 - 6.2.8. Multiple Inheritance Issues**
 - 6.2.9. Static methods in an interface**

7. Exception handling (3 hrs)

Sub Topics

- 7.1. Exception handling fundamentals [Ref 1: Pg.(213)]**
- 7.2. Exception types [Ref 1: Pg.(214)]**
- 7.3. Uncaught exceptions [Ref 1: Pg.(215)]**
- 7.4. Using exceptions**
 - 7.4.1. Try [Ref 1: Pg.(216 – 222)]**
 - 7.4.1.1. Nested try statements [Ref 1: Pg.(220)]**
 - 7.4.2. Catch [Ref 1: Pg.(216 – 222)]**
 - 7.4.2.1. Multiple catch clauses [Ref 1: Pg.(218)]**
 - 7.4.2.1 Multi-catch features [Ref 1: Pg.(231-232)]**
 - 7.4.3. Throw [Ref 1: Pg.(222)]**
 - 7.4.4. Throws [Ref 1: Pg.(223)]**
 - 7.4.5. Finally [Ref 1: Pg.(224)]**
 - 7.4.6. Java's built in Exceptions [Ref 1: Pg.(226)]**
 - 7.4.7. Creating one's own exception sub classes [Ref 1: Pg.(227)]**
 - 7.4.8. Chained exceptions [Ref 1: Pg.(230)]**

Topic 8 : Enumerations, Auto boxing and annotations (Meta data) (2 Hours)

Sub Topics

- 8.1. Enumerations**
 - 8.1.1. Enumeration fundamentals [Ref 1: Pg.(263 – 265)]**
 - 8.1.2. The values() and valueOf() methods [Ref 1: Pg.(266 – 267)]**
 - 8.1.3 Java enumerations are class type [Ref 1: Pg.(267 – 268)]**
 - 8.1.4 Enumerations inherits enum [Ref 1: Pg.(269 – 272)]**

- 8.2. Type wrappers [Ref 1: Pg.(272 – 274)]
- 8.3. Autoboxing
 - 8.3.1. Autoboxing and methods [Ref 1: Pg.(274 – 276)]
 - 8.3.2. Autoboxing/ unboxing occurs in expressions and methods [Ref 1: Pg.(276 – 277)]
 - 8.3.3. Boolean and character values [Ref 1: Pg.(278)]
 - 8.3.4. Autoboxing and preventing errors [Ref 1: Pg.(278 – 279)]
- 8.4. Annotations
 - 8.4.1. Basics [Ref 1: Pg.(279 – 280)]
 - 8.4.2. Specifying a retention policy [Ref 1: Pg.(281)]
 - 8.4.3. Obtaining annotations at run time by use of reflection [Ref 1: Pg.(281 – 284)]
 - 8.4.4. Obtaining all annotations [Ref 1: Pg.(285 – 286)]
 - 8.4.5. The annotatedElement interface [Ref 1: Pg.(286 – 287)]
 - 8.4.6. Using default values [Ref 1: Pg.(287 – 288)]
 - 8.4.7. Marker annotations [Ref 1: Pg.(288 – 289)]
 - 8.4.8. Single-member annotation [Ref 1: Pg.(289 – 290)]
 - 8.4.9. Built in annotations [Ref 1: Pg.(290 – 292)]
 - 8.4.5. Type annotations [Ref 1: Pg.(292 – 297)]
 - 8.4.6. Repeating annotations and restrictions on annotations [Ref 1: Pg.(297 – 299)]

Topic 9 : Understanding generics (1 Hours)

Sub Topics

- 9.1. What are generics [Ref 1: Pg.(337 – 342)]
- 9.2. Generics work only with references types [Ref 1: Pg.(342)]
- 9.3. Generic types differ based on their type arguments [Ref 1: Pg.(342)]
- 9.4. How generics improve type safety [Ref 1: Pg.(342 – 344)]
- 9.5. A generic class with two type parameters [Ref 1: Pg.(345 – 346)]
- 9.6. Generic restrictions [Ref 1: Pg.(377 – 379)]

Topic 10: Overview of some classes of the Java library (16 Hours)

- 10.1. String class and String buffer class [Ref 1: Pg.(413 – 439)]
 - 10.1.1. The string constructors
 - 10.1.2. String length
 - 10.1.3. Special string operations
 - 10.1.4. Character extractions
 - 10.1.5. String comparisons
 - 10.1.6. Searching a string
 - 10.1.7. Modifying a string
 - 10.1.8. Data conversion using valueOf()
 - 10.1.9. Changing the case of a character within a string
 - 10.1.10. Additional string methods
 - 10.1.11. String buffer
 - 10.1.12. String builder
- 10.2. Basics of collection framework and Scanner class
 - 10.2.1. Using Scanner class [Ref 1: Pg.(620 – 628)]
 - 10.2.2. Collection framework
 - 10.2.2.1. Collection overview [Ref 1: Pg.(497 – 500)]

	10.2.2.2.	The collection interfaces [Ref 1: Pg.(501 – 510)]
	10.2.2.3.	The collection classes [Ref 1: Pg.(510 – 521)]
	10.2.2.4.	Accessing a collection via an iterator [Ref 1: Pg.(521 – 525)]
10.3.		Java input and output basics in java.io package
	10.3.1.	Input and output classes, interfaces, files and directories [Ref 1: Pg.(641 – 646)]
	10.3.2.	I/O exceptions and two ways to close a stream [Ref 1: Pg.(649 – 650)]
	10.3.3.	The Byte stream – FileInputStream and FileOutputStream [Ref 1: Pg.(651 – 656)]
	10.3.4.	The character stream – FileReader and FileWriter [Ref 1: Pg.(670 – 674)]
	10.3.5.	The character stream – BufferedReader and PrintWriter [Ref 1: Pg.(676 – 677), (679 – 680)]
	10.3.6.	Serialization [Ref 1: Pg.(682 – 688)]
10.4.		Applets in Java
	10.4.1.	Two types of applets [Ref 1: Pg.(747)]
	10.4.2.	Applet basics [Ref 1: Pg.(748 – 749)]
	10.4.3.	The applet class [Ref 1: Pg.(749 – 750)]
	10.4.4.	Applet architecture [Ref 1: Pg.(751)]
	10.4.5.	Applet skeleton [Ref 1: Pg.(751 – 752)]
	10.4.8.	The HTML applet tag [Ref 1: Pg.(760)]
10.5.		Event handling
	10.5.1.	Event handling mechanism [Ref 1: Pg.(769 – 771)]
	10.5.2.	Event classes [Ref 1: Pg.(771 – 781)]
	10.5.3.	Sources of event [Ref 1: Pg.(780 – 782)]
	10.5.4.	Event listener interfaces [Ref 1: Pg.(782 – 785)]
	10.5.5.	Using the delegation event model [Ref 1: Pg.(785 – 791)]
	10.5.6.	Understanding adapter classes [Ref 1: Pg.(791 – 793)]
	10.5.7.	Inner classes in action [Ref 1: Pg.(793 – 795)]
10.6.		Multithreaded programming
	10.6.1.	Introduction to multithreading [Ref 1: Pg.(233 – 236)]
	10.6.2.	The thread class and runnable interface [Ref 1: Pg.(236)]
	10.6.3.	Main thread [Ref 1: Pg.(237 – 238)]
	10.6.4.	Creating thread [Ref 1: Pg.(238 – 243)]
	10.6.5.	Using isAlive() and join() methods [Ref 1: Pg.(243 – 246)]
	10.6.6.	Thread priorities [Ref 1: Pg.(246)]
	10.6.7.	Synchronization [Ref 1: Pg.(247 – 250)]
	10.6.8.	Interthread communication [Ref 1: Pg.(251 – 255)]
	10.6.9.	Deadlocks [Ref 1: Pg.(255 – 257)]
	10.6.10.	Suspending, resuming and stopping threads [Ref 1: Pg.(257 – 259)]
	10.6.11.	Obtaining thread states [Ref 1: Pg.(259 – 261)]
10.7.		Graphical User Interfaces
	10.7.1.	AWT
	10.7.1.1.	Fundamentals of AWT [Ref 1: Pg.(797 – 810)]
	10.7.1.2.	Displaying information within a window [Ref 1: Pg.(811 – 832)]
	10.7.1.2.	AWT controls fundamentals [Ref 1: Pg.(834 – 855)]
	10.7.1.2.	Layout Managers [Ref 1: Pg.(855 – 870)]
	10.7.1.5.	Menu bars and Menus [Ref 1: Pg.(870 – 876)]
	10.7.1.6.	Dialog boxes [Ref 1: Pg.(876 – 880)]
	10.7.2.	Swing

10.7.2.1.	Introducing Swings [Ref 1: Pg.(1021 – 1024)]
10.7.2.2.	Components and containers [Ref 1: Pg.(1024 – 1025)]
10.7.2.3.	The Swing packages [Ref 1: Pg.(1026)]
10.7.2.4.	Event handling in Swings [Ref 1: Pg.(1030 – 1035)]
10.7.2.5.	Swings components at work [Ref 1: Pg.(1041 – 1068)]
10.7.2.6.	Creating a swing menus [Ref 1: Pg.(1069 – 1086)]
10.7.2.7.	Toolbars in Swings [Ref 1: Pg.(1087 – 1094)]
10.7.3	JavaFX
10.7.3.1.	JavaFx basics [Ref 1: Pg.(1105 – 1108)]
10.7.3.2.	JavaFx application skeleton [Ref 1: Pg.(1108 – 1111)]
10.7.3.3.	Compiling and running JavaFx applications [Ref 1: Pg.(1111 – 1112)]
10.7.3.4.	JavaFx controls and events [Ref 1: Pg.(1112 – 1119)]
10.7.3.5.	JavaFX controls [Ref 1: Pg.(1125 – 1165)]
10.7.3.6.	Effects and transforms [Ref 1: Pg.(1165 – 1170)]
10.8	Introduction to database connection
Teaching /Learning Methods: You can access all learning materials and this syllabus in the VLE: http://vle.bit.lk , if you are a registered student of BIT degree program. It is very important to participate in learning activities given in the VLE to learn this subject.	
Assessment Strategy: Continuous Assessments/Assignments: The assignments consist of two quizzes, assignment quiz 1 (It covers the first half of the syllabus) and assignment quiz 2 (It covers the second half of the syllabus). The maximum mark for a question is 10 and the minimum mark for a question is 0 (irrespective of negative scores). Final assignment mark is calculated considering both assignments, and students will have to obtain at least 40% for each assignment. Students are advised to complete online assignments before the given deadline. It is compulsory to pass all online assignments to qualify to obtain the Level I, Diploma in IT (DIT), certificate. In the course, case studies/Lab sheets will be introduced, and students have to participate in the learning activities.	
Final Exam: The final examination of the course will be held at the end of the semester. The paper consists of 40 MCQs and candidates have to answer all the 40 questions within 2 hours.	
References/ Reading Materials: <ul style="list-style-type: none"> • Ref 1: Java: The Complete Reference TM, ninth Edition by Herbert Schildt, Tata McGraw-Hill Edition 2014 • Supplementary Reading: <ol style="list-style-type: none"> 1. Java SE Development Kit 8 Downloads https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html 2. WAMP server for MySql database 	