COURSE: DFC3013 OBJECT ORIENTED PROGRAMMING

CREDIT(S): 3

PRE REQUISITE(S): DFC1023 PROBLEM SOLVING AND PROGRAM DESIGN

SYNOPSIS

OBJECT ORIENTED PROGRAMMING course introduces students to the knowledge of object-oriented technology to equip students with the concept of object oriented programming using Java language. This course introduces students to write, compile and run Java programs, make effective use of some of the standard packages, write object-oriented code using classes and objects, inheritance and polymorphism.

COURSE LEARNING OUTCOMES (CLO)

Upon completion of this course, students should be able to:

1. explain the concepts of object oriented design, methodology and programming in application development. (C2, PLO1)

2. apply Object Oriented Concepts in designing a successful program using programming tools. (P3, C3, PLO2, PLO1)

3. solve problems using the Object Oriented Programming approach and exception handling to produce well engineered program. (P3, A3, PLO2, PLO4)
### SUMMARY (LECTURE: PRACTICAL)

<table>
<thead>
<tr>
<th>SST</th>
<th>RTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>06 : 06</td>
</tr>
<tr>
<td><strong>INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING (OOP)</strong></td>
<td></td>
</tr>
<tr>
<td>This topic provides an introduction to OOP. It explains the history, concept and the importance of OOP. This topic describe the general UML elements method in problem solving. This topic also develops an object-oriented way of thinking and introduce Unified Modelling Language (UML) – a graphical language that allows people who design object-oriented software system to use an industry standard notation to represent them.</td>
<td></td>
</tr>
</tbody>
</table>

| 2.0       | 08 : 08 |
| **FUNDAMENTALS OF THE JAVA PROGRAMMING LANGUAGE** | |
| This topic discusses on evolution of Java, terminology and environment, features of Java and Java technology. It focuses on the use of variables and operators, as well as the use of Input And Output Stream, branching statements and arrays. | |

| 3.0       | 12 : 12 |
| **CLASSES, INHERITANCE AND POLYMORPHISM** | |
| This topic explains the concept of inheritance, class and polymorphism. This topic also introduces and explains the concepts of abstraction and polymorphism. The students are also exposed to the use of package and string in Java programs. | |

| 4.0       | 04 : 04 |
| **MULTI THREADING AND EXCEPTION HANDLING** | |
| This topic introduces the concept of multiple threads, an important feature that allows multiple programs to run concurrently. It also covers the error handling facilities and the importance of exception handling mechanism built into the Java programming language. | |

### DEPENDENT LEARNING COURSEWORK ASSESSMENT (02)

**RTA** – Recommended Time Allocation

**SST** – Suggested Sequence of Topics
SYLLABUS

1.0 INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING (OOP)

1.1 Understand the concepts of Object-Oriented Programming.
   1.1.1 Define OOP concept.
   1.1.2 Describe the history of OOP.
   1.1.3 List the advantages of using OOP.
   1.1.4 Illustrate the basic terminologies of OOP: Classes, Object, Encapsulation, Data Abstraction, Inheritance, and Polymorphism.
   1.1.5 Distinguish between abstraction and encapsulation
   1.1.6 Differentiate between Structured Programming and OOP approach.

1.2 Understand basic object technology concept and terminology.
   1.2.1 Define object, attribute and behaviour.
   1.2.2 Explain Object-oriented Analysis and Design (OOAD) method in problem solving.
   1.2.3 Understand the fundamental of Unified Modelling Language (UML).
   1.2.4 Describe general UML elements.
   1.2.5 Design classes using UML Class Diagram.
   1.2.6 Identify components of a class:
      a. structuring classes
      b. class declaration
      c. variable declaration and assignments
      d. comments
      e. methods

2.0 FUNDAMENTALS OF THE JAVA PROGRAMMING LANGUAGE

2.1 Understand Java terminology and environment
   2.1.1 Describe the history of Java technology.
   2.1.2 Identify the features of Java programming language.
   2.1.3 State anatomy of the Java programs:
      a. comment
      b. reserved word
      c. modifiers
      d. statements
      e. blocks
      f. classes
      g. method
      h. main method

   2.1.4 Apply programming style and documentation in Java:
      a. appropriate comments and comments style
      b. naming conventions
      c. proper indentation and spacing
      d. block styles

   2.1.5 Write a simple Java programs.
   2.1.8 Compile and run the Java programs.
2.1.9 Troubleshoot the source code to identify the following errors:
   a. compile-time errors  
   b. runtime errors  
   c. logic errors  

2.2 Use variables, operators and Input/Output Streams.
2.2.1 Declare, initialize, and use variables and constants according to Java programming language.
2.2.2 Identify the mathematical, relational and conditional operators in Java Programming. Explain type casting in Java programs.
2.2.3 Implement type casting to change the data types.
2.2.4 Define input stream (System.in) and output stream (System.out) in Java programs.
2.2.5 Implement input stream (System.in) and output stream (System.out) in Java programs.
2.2.6 Write java programs using variables, operators and Input/Output Streams.

2.3 Use branching statements and arrays.
2.3.1 Explain selection statements in Java programs.
2.3.2 Explain looping statements in Java programs.
2.3.3 Write program using branching statements.
2.3.4 Define an array in java program.
2.3.5 Create reference arrays of objects in Java program.
2.3.6 Initialize elements of arrays.
2.3.7 Pass and return array to method.
2.3.8 Write program using single and multidimensional array.

3.0 CLASSES, INHERITANCE AND POLYMORPHISM

3.1 Design Class.
3.1.1 Identify built-in classes in Java library.
3.1.2 Define method in Java programs.
3.1.3 Identify the use of methods in Java programs.
3.1.4 Write methods in Java programs.
3.1.5 Create an object in Java programs.
3.1.6 Explain the concepts of accessing objects.
3.1.7 Explain the scope of variables in classes: instance and local variables.
3.1.8 Pass parameters by value to methods in Java programs.
3.1.9 Define constructor and constructor overloading in Java programs.
3.1.10 Implement constructor and constructor overloading in Java programs.
3.1.11 Define method overloading in Java programs.
3.1.12 Implement method overloading in Java programs.
3.1.13 Explain data field encapsulation in Java programs.
3.1.14 Pass objects to method.
3.1.15 Modify the behaviour of classes, methods, variables of constructor using modifiers.

3.2 Apply String in Java programs.
3.2.1 Explain String references as parameters.
3.2.2 Differentiate between the String and StringBuffer class.
3.2.3 Write Java programs using String and StringBuffer objects.
3.2.4 Convert a String to a primitive data in Java programs.
3.2.5 Construct program using method in String class.

3.3 Create Packages in Java Programs
3.3.1 Define and identify packages and it uses.
3.3.2 Explain system packages in Java programs.
3.3.3 Explain naming conventions for packages.
3.3.4 Create packages in Java programs.

3.4 Apply Inheritance in Java programs.
3.4.1 Describe inheritance in Java programs.
3.4.2 Apply keyword super in Java programs.
3.4.3 Explain overriding method in Java programs.
3.4.4 Implement overriding method in Java programs.

3.5 Implement Polymorphism in Java programs
3.5.1 Explain polymorphism in Java programs.
3.5.2 Implement polymorphism in Java programs.

3.6 Implement Abstract Classes in Java programs.
3.6.1 Explain abstract classes in Java programs.
3.6.2 Implement abstract classes in Java programs.

3.7 Apply Interface in Java programs.
3.7.1 Explain interface in Java programs.
3.7.2 Classify the built-in interface class in Java library.
3.7.3 Create Java program using the following classes:
   a. extending interface
   b. implementing interface

3.8 Implement Abstract Methods in Java programs.
3.8.1 Explain abstract methods in Java programs.
3.8.2 Implement abstract methods in Java programs.

4.0 MULTI THREADING AND EXCEPTION HANDLING

4.1 Understand the concept of Threading.
4.1.1 Define Thread and its uses in Java program.
4.1.2 Explain the different types of Thread:
   a. Single thread
   b. Multiple thread

4.1.3 Differentiate between multitasking and multithreading.
4.1.4 List the methods involved in the life cycle of a thread.
4.1.5 Write multithreaded application.

4.2 Understand concept of exception handling.
4.2.1 Define the concept of exception handling mechanism.
4.2.2 Explain the use of exception handling.
4.2.3 Explain the different types of exceptions in RuntimeException
   a. NumberFormatException
   b. ArrayIndexOutOfBoundsException
   c. ArithmeticException

4.2.4 Create Java programs using exception handling.
REFERENCES

Main :


Additional :


# DFC3013 Object Oriented Programming

## MATRIX OF COURSE LEARNING OUTCOMES (CLO) VS PROGRAMME LEARNING OUTCOMES (PLO)

<table>
<thead>
<tr>
<th>Course Learning Outcome (CLO)</th>
<th>Compliance to PLO</th>
<th>Recommended Delivery Methods</th>
<th>Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PLO1</td>
<td>PLO2</td>
<td>PLO3</td>
</tr>
<tr>
<td>1. Explain the concepts of object oriented design, methodology and programming in application development.</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Apply Object Oriented Concepts in designing a successful program using programming tools.</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C3</td>
<td>P3</td>
<td></td>
</tr>
<tr>
<td>3. Solve problems using the Object Oriented Programming approach and exception handling to produce well engineered program.</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

**Remark:**
LD 1 Knowledge
LD 2 Practical Skills
LD 3 Communication Skills
LD 4 Critical Thinking and Problem Solving Skills
LD 5 Social Skills and Responsibilities
LD 6 Continuous Learning and Information Management Skills
LD 7 Management and Entrepreneurial Skills
LD 8 Professionalism, Ethics and Moral
LD 9 Leadership and Teamwork Skills
ASSESSMENT
The course assessment comprises two components namely:

i. **Coursework Assessment (CA) – 50%**
   Coursework assessments that measures knowledge, practical skills and generic skills are carried out in the form of continuous assessment. Coursework assessments total score comprises the knowledge and practical marks **ONLY**. It does not include the mark of generic skills.

ii. **Final Examination Assessment (FE) – 50%**
   Final examination is carried out at the end of the semester.

### ASSESSMENT SPECIFICATION TABLE (AST)

<table>
<thead>
<tr>
<th>COURSE LEARNING OUTCOMES (CLO)</th>
<th>TOPICS</th>
<th>ASSESSMENT METHODS FOR COURSEWORK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1. Explain the concepts of object oriented design, methodology and programming in application development. (C2, PLO1)</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>2. Apply Object Oriented Concepts in designing a successful program using programming tools. (P3, C3, PLO2, PLO1)</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>3. Solve problems using the Object Oriented Programming approach and exception handling to produce well engineered program. (P3, A3, PLO2, PLO4)</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

**Remark**
- Topic 1: Introduction to Object-Oriented Programming (OOP)  
  Refers to the CLO to be assessed through the indicated assessment task.
- Topic 2: Fundamentals of Java Programming Language  
  *(#)*  
  # refers to the quantity of assessment
- Topic 3: Classes, Inheritance and Polymorphism  
  Indicates the topic (s) to be covered under the assigned/identified assessment tasks. For merged topics, lecturers have the options of choosing the preferred topic (s).
- Topic 4: Multi Threading and Exception Handling  
  **  
  The generic skills are to be assessed separately. The total score for generic skills is **100%**. However, it is **NOT PART** of the coursework assessment mark.
## DISTRIBUTION OF STUDENT LEARNING TIME (SLT) 
### ACCORDING TO COURSE LEARNING - TEACHING ACTIVITY

<table>
<thead>
<tr>
<th>No.</th>
<th>Learning and Teaching Activity</th>
<th>SLT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DEPENDENT LEARNING</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>Delivery Method</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>Lecture</td>
<td>30</td>
</tr>
<tr>
<td>1.2</td>
<td>Practical</td>
<td>30</td>
</tr>
<tr>
<td>1.3</td>
<td>Tutorial</td>
<td>0</td>
</tr>
<tr>
<td>2.0</td>
<td>Coursework Assessment (CA)</td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>Lecture-hour-assessment</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Test</td>
<td>1</td>
</tr>
<tr>
<td>-</td>
<td>Quiz</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Practical-hour-assessment</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Laboratory Exercises</td>
<td>3</td>
</tr>
<tr>
<td>2.3</td>
<td>Tutorial-hour-assessment</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td><strong>INDEPENDENT LEARNING</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>Coursework Assessment (CA)</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Problem Based Task</td>
<td>1</td>
</tr>
<tr>
<td>4.0</td>
<td>Preparation and Review</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Lecture</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Preparation before theory class eg: download lesson notes.</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Review after theory class eg: additional references, discussion group,discussion</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Preparation for quizzes.</td>
<td>30</td>
</tr>
<tr>
<td>4.2</td>
<td>Practical</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Preparation before practical class/field work /survey eg: review notes, check list/ labsheets and/or tools and equipment.</td>
<td>15</td>
</tr>
<tr>
<td>-</td>
<td>Post practical activity eg: lab report, additional references and discussion session</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Preparation before studio work presentation/critique.</td>
<td></td>
</tr>
<tr>
<td>4.3</td>
<td>Tutorial</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Preparation for tutorial</td>
<td>0</td>
</tr>
<tr>
<td>4.4</td>
<td>Assessment</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>Preparation for Test.</td>
<td>1</td>
</tr>
<tr>
<td>-</td>
<td>Preparation for final examination.</td>
<td>2</td>
</tr>
<tr>
<td><strong>FINAL EXAMINATION</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>Credit = SLT/40</td>
<td>3</td>
</tr>
</tbody>
</table>

**Remark:**

1. Suggested time for
   
   - Quiz : 10 - 15 minutes
   - Test (Theory) : 30 - 60 minutes
2. Laboratory Exercises are conducted during practical hours
3. 40 hours is equivalent to 1 credit