

### **CMP2103 Objected Oriented Programming**

Period per Week			Contact Hour per Semester	Weighted Total Mark	Weighted Exam Mark	Weighted Continuous Assessment Mark	Credit Units
LH	PH	TH	CH	WTM	WE	WCM	CU
45	30	00	60	100	60	40	4

#### **Rationale**

In the object-oriented view of software, programs are considered to be collections of objects that interact by sending messages to one another and reacting to the answers to those messages. These ideas are at the forefront of modern software development. This course is designed to teach the fundamental ideas behind the object-oriented approach to programming; through the widely used Java programming language. The course concentrates on those aspects of the Java language that best demonstrate object-oriented principles and good practice. CMP2104 will give the student a solid basis for further study of the Java language and object-oriented software development.

## Objectives

- To build the students' skills geared towards professional Object Design and Programming with Java
- To introduce the students to how Object-Orientation works and how to best apply it in Java
- To develop the students' mastery in the areas of Inheritance, Encapsulation, Abstraction & Polymorphism in Java
- To provide students with extensive hands-on experience with Java programming
- To impart Exception Handling techniques mechanism
- To familiarize the students with the methods for manipulation of Java Object Collections

## Course Content

### 3. *History and Overview of Objected-Oriented-Programming*

- Definition, Concept of Object-Orientedness
- History and Evolution of Objected-Oriented-Programming
- Object-Oriented-Programming Languages(Classification, Examples)
- Key Principles of an Object-Oriented-Programming Language
- Comparative Analysis of popular Object-Oriented-Programming Languages (Java, C++ and C#)

### 4. *Platforms for Object-Oriented-Programming*

- The concept of a Computing Platform
- Java Platforms (Smart Card Edition, Standard Edition, Micro-Edition, Enterprise Edition)
- Open Source Platforms (PHP, Python)
- .NET Platform

### 5. *Fundamental Programming Structures in Java*

- Primitive Data Types
- Variables
- Constants
- Assignments
- Initializations
- Operators
- Strings
- Control Flow

### 6. *Classes and Objects in Java*

- Classes & Objects
- Object Oriented Programming Principles(Instantiation, Encapsulation, Specialization)
- Instance Variables
- Class Variables
- Constructors
- Instance Methods
- Class Methods
- Method Overloading
- The this keyword
- Passing and returning objects
- Garbage Collection in Java

### 7. *Object Design and Programming with Java*

- Abstraction

- Inheritance
  - Polymorphism
  - Method Overriding
  - Associations
  - Delegation
8. **Java Interfaces**
- Creating high levels of abstraction
  - Purpose of Interfaces
  - When to use them
  - Interface Declaration
  - Implementing an Interface
  - Interface Inheritance
9. **Java Exception Handling**
- Why Exceptions
  - Standard Exception Handling Options
  - Exception Class Hierarchy
  - Checked vs. Unchecked Exceptions
  - Catching an Exception: try and catch blocks
  - Methods Which Throw Exceptions: the throws clause
  - Handling vs. Declaring Exceptions
  - System Exceptions vs. Application Exceptions
  - Writing Custom Exceptions
10. **Java Collections API**
- Arrays
  - The Java Collections Framework
  - Collections Interfaces
  - Concrete Collections
  - Iterating through Collections
11. **Java Input/Output API**
- Streams & Files
  - Input & Output Streams
  - File Streams
  - Object Streams
  - Object Serialization
  - Readers & Writers

### **Learning Outcomes**

- Knowledge and Understanding Upon successful completion of the module, a student will: Understand basic principles of object-oriented program design. Understand the basic and some advanced issues related to writing classes and methods - such as data, visibility, scope, method parameters, object references, and nested classes. Understand the basic ideas behind class hierarchies, polymorphism, and programming to interfaces. Get exposure to exceptions and basic I/O streams. Understand basic principles, main features and operations of abstract data types, in particular of lists, stacks, queues, trees, heaps, hash tables and graphs. Differentiate specifications of abstract data types from particular implementation techniques. Learn about fundamental algorithms associated with the above data types, including tree traversal, treesort, heapsort and graph traversal algorithms.

- Intellectual and Practical skills Upon successful completion of the module, a student will: Be able to solve a given application problem by going through the basic steps of program specifications, analysis, design, implementation and testing --- within the context of the object-oriented paradigm. Be able to competently read 'foreign' Java source code and object diagrams. Have developed solid Java programming skills and the ability to put in practice the acquired knowledge and understanding of the Java language and object-oriented design in relatively simple case studies. Be able to develop Java implementations of abstract data types using different approaches, and evaluate their differences. Be able to use abstract data types and related implementations in designing and implementing efficient solutions to straightforward application problems.

**Recommended and Reference Books**

- [1] Timothy Budd, *Understanding Object-Oriented Programming with Java*, 2nd Edition Addison-Wesley Longman, 1999, ISBN: 0-201-61273-9,
- [2] Y. Daniel Liang, *Introduction to Programming with C++*, Prentice Hall, 2007
- [3] Schach Stephen, *Object-Oriented and Classical Software Engineering*, 7<sup>th</sup> Edition, 2006, McGraw-Hill. ISBN 0-073-19126-4.
- [4] Bruce E. Wampler, *The Essence of Object-Oriented Programming*, Addison-Wesley, 2001.