<table>
<tr><td>1.</td><td>Name of Course</td><td colspan="6">Advanced Java Programming</td></tr>
<tr><td>2.</td><td>Course Code</td><td colspan="6">CNET3563</td></tr>
<tr><td>3.</td><td>Name(s) of academic staff</td><td colspan="6"></td></tr>
<tr><td>4.</td><td>Rationale for the inclusion of the course/module in the programme</td><td colspan="6"><u>Elective</u>
This module develops problem-solving and programming skills essential in professional programming using the Java language. By taking this module students will learn how to write well-structured object oriented programs using Java. It also introduce various programming paradigms, and programming methodology</td></tr>
<tr><td>5.</td><td>Semester and Year offered</td><td colspan="6">2/3</td></tr>
<tr><td rowspan="3">6.</td><td rowspan="3">Total Student Learning Time (SLT)</td><td colspan="4">Face to Face</td><td colspan="2">Total Guided and Independent Learning</td></tr>
<tr><td rowspan="2">L = Lecture
T = Tutorial
P = Practical
O= Others</td><td>L</td><td>T</td><td>P</td><td>O</td><td rowspan="2">Independent = 84
Total =140</td></tr>
<tr><td>28</td><td></td><td>28</td><td>84</td></tr>
<tr><td>7.</td><td>Credit Value</td><td colspan="6">3</td></tr>
<tr><td>8.</td><td>Prerequisite (if any)</td><td colspan="6"><u>None</u></td></tr>
<tr><td>9.</td><td colspan="7">Objectives:
Object Oriented Programming has become the predominant technique for writing software in the past decade. Many other important software development techniques (e.g. modern distributed systems and component technology) are based upon the fundamental ideas captured by object-oriented programming. This course will introduce various programming paradigms, programming methodology and show students how to write well-structured object oriented programs using Java.</td></tr>
<tr><td>10.</td><td colspan="7">Learning outcomes:
By the end of the subject, students should be able to:
• Develop and apply algorithms for various programming paradigm and methodology.
• Demonstrate the implementation of object oriented programming concepts and design using a high-level programming language, like Java correctly and effectively.</td></tr>
<tr><td>11.</td><td colspan="7">Transferable Skills:
• Become skilled in the use of java programming languages and set up threads that work without errors.
• Describe the pros and cons of the Java programming language.
• Write well-structured object oriented programs using Java.</td></tr>
<tr><td>12.</td><td colspan="7">Teaching-learning and assessment strategy
A variety of teaching and learning strategies are used throughout the course, including:
• Classroom lessons. Lectures and Power Point presentations
• Laboratory sessions: Practice exercises
• brainstorming;
• Lecturer-led problem-solving sessions,
• Solving assigned problems in groups and singly
• collaborative and co-operative learning;
• Independent study.

Assessment strategies include the following:
• Ongoing quizzes
• Midterm tests
• Performance Assessment (Project, Assigned exercises)
• Lecturer Observation</td></tr>
</table>

13. **Synopsis:**

This module develops problem-solving and programming skills essential in professional programming using the Java language. The major areas of study include: Comparison of procedure oriented, structure oriented and object oriented programming paradigms, Top-down design, algorithm development, Fundamentals of object-oriented design, Classes, Attributes and Behaviour, Java fundamentals, Data members and member functions, Dynamic memory allocation, Concept of inheritance and polymorphism, Advanced Java concepts and applications

14. **Mode of Delivery:**

- Classroom lessons. Lectures and Power Point presentations
- Laboratory sessions: Practice exercises

15. **Assessment Methods and Types:**

The assessment for this course will be based on the following:

Coursework **50%**

- Quizzes 10%
- Project and assignments 20%
- Mid-Semester Exam 20%

Final Examination **50%**

100%

16. **Mapping of the course/module to the Programme Aims**

A1	A2	A3	A4	A5	A6	A7	A8	A9
4	2	2	1	3	1	1	2	0

17. **Mapping of the course/module to the Programme Learning Outcomes**

LO1	LO2	LO3	LO4	LO5	LO6	LO7	LO8	LO9	LO10	LO11	LO12
4	2	2	0	3	0	1	2	0	1	0	0

18. **Content outline of the course/module and the SLT per topic**

	Details	SLT			
		L	P	Indep.	Total
Topic 1	**Introduction to Java** Object-Oriented Programming, Java terminology, Programming Concepts & Structures, Introducing, Type conversions, fundamental of object oriented design - identifying classes - attributes and behaviour - features of object oriented programming	4	4	12	**20**
Topic 2	**Java Fundamentals** Standard I/O streams - function prototypes – JAVA enhancements to C++ – Default function parameters - inline functions – overloaded functions - reference variables comparison between pointers and references.	4	4	12	**20**
Topic 3	**Classes** Creating new data type in JAVA- class declaration - members - constructors and destructors - access functions constant objects - member objects - static members - friend classes - arrays of class objects.	4	4	12	**20**
Topic 4	**Dynamic Memory Allocation** Free store - new and delete operators - class with pointer members - this pointer assignment - initialization - copy constructor - passing and returning objects - advanced free store techniques – exception handling.	4	4	12	**20**

	Topic 5	**Inheritance and Polymorphism** Operator overloading - handling related types in JAVA- derived class – conversion between base and derived classes – virtual functions - dynamic binding - pure virtual functions - protected members – public and private base classes - new, delete operators overloading - inheritance applications.	6	6	18	**30**
	Topic 6	**Advanced JAVA concepts an applications** File handling - templates - container classes – class library - stack, queue and linked list applications - simple database applications.	4	4	12	**20**
	Topic 7	**JavaScript** Placing JavaScript code in HTML pages Using comments	2	2	6	**10**
		Total hours	28	28	84	140
	Laboratory	Laboratory Details Exercises based on topics covered in each lecture. programming work must include the following • Type conversions, attributes and behaviour • identifying classes • Default function, inline functions, overloaded functions, reference variables, pointers and references. • Class declaration, member objects - static members - friend classes - arrays of class objects. • constructors and destructors • Dynamic Memory Allocation, Free store, new and delete operators, class with pointer members • copy constructor, passing and returning objects • exception handling • virtual functions • dynamic binding • pure virtual functions - protected members – public and private base classes - new, delete operators overloading - inheritance applications. • I/O • File handling - templates - container classes – class library - stack, queue and linked list applications - simple database applications.				
19.	**Main references supporting the course:** *Y. Daniel Liang,* ***Introduction to Java Programming, Comprehensive Version*** *(7th Edition) (2008)*					
	Additional references supporting the course: 1. *David J. Barnes and Michael Kolling,* ***Objects First With Java: A Practical Introduction Using BlueJ*** *(4th Edition) (Paperback - Sep 1, 2008)* 2. *Herbert Schildt,* ***Java: The Complete Reference****, Seventh Edition (Complete Reference Series) (2006)* 3. *Cay S. Horstmann,* ***Big Java*** *(2007)*					
20.	**Other additional information** All materials will be available to the students online.					