TECHNICAL UNIVERSITY OF KENYA
Education and Training for the Real World

School of Computing and Information Technologies

ECSI 3205 / ECII 3205 / ECCI 3205
Object oriented Programming (OOP)

Course outline and Course Summary

Prof. Salesio M. Kiura (salesio.kiura - @gmail.com / @tukenya.ac.ke) 0720- / 0780- 370071

Prerequisite

- Introduction to Programming, Structured Programming, C or C++ (or any other programming Language)

Purpose of the Course

The course introduces the students to the concepts and principles of Object Oriented Programming. The central theme will be about using object orientation in coming up with software with an emphasis on developing insights about how object orientation changes the way we conceptualize, design, develop and implement computer systems

Learning Outcomes

At the end of this course unit, the learner should be able to:

1. Demonstrate an in-depth understanding of Object Oriented paradigm and concepts
2. Apply object oriented concepts using a selected language (Java)
3. Implement principles of inheritance, exception handling, abstract classes, packages, etc
4. Analyze application scenarios (for) and design software systems using object oriented analysis and design.

Delivery Method:

Facilitated lectures, Practical sessions, Problem Based Learning (individual and group presentations), Tutorials, Independent study. The course will also have an online page. Students are encouraged to actively participate and engage online as well.

Instructional materials and equipment:

A computer installed with an IDE such as Netbeans, IntelliJ IDEA, JCreator, Eclipse (OR any other IDE) and Java Development Kit (JDK) 1.7 or later. These IDEs and many other tools are free for download online. In this course the lecturer will make demos using Netbeans

Register by following the link:

https://elearning.tukenya.ac.ke

Site Home >> Fac. Of Applied Sciences >> Computing and Information Technologies >> Degree Courses … >> …appropriate grouping…

Couse Content and plan outline

	Topic	Sub topics / Coverage
1.	Introduction	- Introductions and course expectations - Define OOP as a concept, as an approach - How did it evolve - OOP paradigms - OOP languages - Comparison with structured programming - Merits and Demerits of OOP
2.	OOP Concepts ➔	- Key OOP concepts (an overview) - OOP Benefits (Strengths: contributions to Software Engineering, good programs, best practices)

3.	Getting started with Java	- Developing a simple Java program
		- Nice things about Java;
		- ENV Set up: Java, JDK, JRE, IDEs
		- Hello World Application
		  o Filenames
		  o Variable types, declarations

4.	OOP Program	- OOP Language structure***?? - (Anatomy of) A Java program - Inter object communication [demonstrate use of a menu class, other classes] - Program control structures –

	Topic	Sub topics / Coverage
5.	Projects	- Groups – the following concepts will be implemented in projects - NB: Takes 20% of the course mark
6.	Objects and Classes Practical implementation	- Definition of Objects and Classes - Implementation of Objects and classes - Importance of objects and classes - Constructors WILL HAVE COME EARLIER IN THE NOTES
7.	Functions Practical implementation	- Introduction - Implementations – declarations - Parameters passing and referencing - Scope WILL HAVE COME EARLIER IN THE NOTES
8.	Overloading Practical implementation	- Functions overloading - Constructor overloading - Operator Overloading WILL HAVE COME EARLIER IN THE NOTES
9.	Inheritance Practical implementation	- Importance - Rules and types - Information hiding [visibility of variables] WILL HAVE COME EARLIER IN THE NOTES

	Topic	Sub topics / Coverage
10.	Polymorphism Practical implementation	- Introductions - Implementation WILL HAVE COME EARLIER IN THE NOTES
11.	File Operations Practical implementation	- Creating and manipulating files - WILL HAVE COME EARLIER IN THE NOTES
12.	Mini-Project	- Problem definition - Scope – statement of functionalities - Design [flow of logic] - Implementation - Documentation