

## **CMP2103 OBJECT ORIENTED PROGRAMMING**

Hours per Semester				Weighted Total Mark	Weighted Exam Mark	Weighted Continuous Assessment Mark	Credit Units
LH	PH	TH	CH	WTM	WEM	WCM	CU
30	60	00	60	100	60	40	4

### **Rationale**

The course helps students to acquire more knowledge in object oriented computer programming techniques, and computer graphics.

### **Course Objectives**

The goal of this course is to study key concepts related to computer programming for scientific and engineering applications. through an advanced view of computer programming, mainly using Java, and C++. It includes a study of the differences and similarities between Java and C++; the use of current operating systems (e.g. Linux and Unix) and compilers (e.g. gcc), apply computer graphics to produce engineering drawings and illustrations, Carry out graphical user interface design as well as details of Object Oriented Programming. Hands on programming should be a key part of the course.

### **Detailed Course Content:**

Brief Introduction to Software Engineering	<b>[ 4 Hours]</b>
Programming in Java:	<b>[ 12 Hours]</b>
C++ vs. Java; Introduction to Java, Java Applications, Java Applets, Control Structures and Arrays, Methods	
Object Oriented Programming:	<b>[ 12 Hours]</b>
Graphics:	<b>[ 8 Hours]</b>
Algorithms for 2d and 3d drawing; Picture manipulation and transformation; curve plotting and mouse interaction; Use of standard packages and graphics within a window environment;	
Graphical User Interfaces:	<b>[ 16 Hours]</b>
Introduction to human/computer interaction, models of user knowledge, dialogue design, data display; error control; prototypes and acceptance testing; designing menu systems	
Exception Handling:	<b>[ 8 Hours]</b>
Threads	
Files and Streams	
Message Passing Interface (MPI and Parallel Computing)	

### **Mode of Delivery**

The course will be taught by using lectures, tutorials and assignments.

### **Assessment**

Assignments, tests, laboratories, and final examination. Their relative contributions to the final grade are :

### **Requirement**

### **Percentage contribution**

Course work (Assignments, laboratories, tests)	40%
Final examination	60%
<b>Total</b>	<b>100%</b>

### **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.

### **Method of Teaching / Delivery**

The course will be taught by using lectures, tutorials and assignments.

### **Mode of Assessment**

Assignments, tests and final examination. Their relative contributions to the final grade are :

<b>Requirement</b>	<b>Percentage contribution</b>
Course work (Assignments, tests)	40%
Final examination	60%
<b>Total</b>	<b>100%</b>

### **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.

### **Possible Lecturers:**

Dr. D. Okello  
 Mr. S. Mwanje  
 Mr. A. Tumwesigye  
 Mr. P. I. Musasizi  
 Mr. P. Serwanga

