Wollo University

Kombolcha Institutes of Technology

College of Informatics

Department of Information System (IS)

Object Oriented Programming

INSY2042

Prepared By Daniel G.

ሰኔ 2013 ዓ.ም

Course Out Line				
Program	Information System (Regular)			
Course Information				
Module Title	***Advanced Programming***			
Module code	INSY-M2042			
Course Title	Object Oriented Programming			
Course Code	INSY2042			
ECTS	7			
Study Hours	**Lecture**	**Lab/Practical**	**Tutorial**	**Home Study**
	Lec. 48	Lab. 48	Tut. 16	HS. 77
Course Type	Core			
Prerequisite				
Academic Year	2013			
Semester	II			
Target Group	**2nd Year Information system regular Students**			
Class Room	CR-1722			
Instructor Information				
Name	Daniel G.			
Office				
Address	E-mail: danielgetaye1543@gmail.com			

Course Description

Students' previous exposure to and skills in structured programming is assumed (although a review of the necessary basic programming notions is included). The course is designed to introduce how to develop business applications using object-oriented design methodology with Java as an illustration programming language. It includes Object Oriented Programming paradigm and its use; classes, Objects, Abstraction and Encapsulation; Inheritance, Polymorphism, Creating Graphical User interfaces (GUIs), Data Structures, Exceptions(Try, catch, and throw, finally how exceptions affect the design of an application), File input/output (I/O), Threads and Java Database Connectivity.

Course Objectives

At the end of the course students will be able to

- understand the principles of software engineering with emphasis on the various phases of the software development life cycle
- Understand major concepts of object-oriented programming
- understand the programming environment as defined by compilers, interpreters, editors, and other system software providing support for the programming activity
- develop skills in OO design and program development within an integrated development environment

- Use arrays and other data structures
- To understand the concepts of encapsulation, inheritance
- Implement I/O functionality to read from and write to data and text files.
- understand object technology and its applications
- Explain the application of a variety of data structures, understand the advantages and disadvantages of those structures
- To be familiar with object-oriented design concepts, and have had practical experience of designing and implementing object-oriented software In general, a student should be competent in:
- design and implement a small-scale object-oriented program with 5-10 classes in the Java language, use a large sub-set of the Java programming language effectively; gain an understanding of how to design and use classes correctly; understand class relationships and implement common data structures.

Course Contents and Schedule

Date	Contact Hours	Topics or Subtopics or Chapters	Learning Outcomes
Week 1	6 Hours	**Chapter 1: Introduction to Object-Oriented Programming (OOP)** 1.1. Overview of OOP? 1.1 Why Java? 1.2 The JVM and Byte Code 1.3 Basic concepts of OOP 1.3.1 Classes 1.3.2 Objects 1.3.3 Members 1.3.4 Class member visibility 1.4 Encapsulation, Inheritance and Polymorphism	After completing this chapter, the students are expected to: ➢ Understand overview OOP and Java ➢ Identify the major concepts of OO ➢ Understand the concept of encapsulation, inheritance and polymorphism

Week 2	**6 Hours**	**Chapter 2: Objects and classes** More on OOP concepts 2.1. Member methods and their components 2.2. Instantiation and initializing class objects 2.3. Constructors 2.3.1. Default and parameterized 2.3.2. Overloaded constructors 2.4. Methods 2.5. Access Specifier 2.6. Accessors and Mutators 2.7. Calling and returning methods 2.8. Static and instance members	After completing this chapter, the students are expected to: ➢ Understand More on OOP concepts ➢ Identify major concepts of constructor ➢ Understand the concept of Methods, Access Specifier, Accessors and Mutators
Week 3	**6 Hour**	**Chapter 3: Inheritance** 3.1. Concept of inheritance 3.2. Super classes and subclasses 3.3. Protected members 3.4. Overriding methods 3.5. Using this() and super() 3.6. Use of final with inheritance Constructors in subclasses	After completing this chapter, the students are expected to: ➢ Understand the concept of Inheritance
Week 4	**7 hours**	**Chapter 4: Polymorphism** 4.1. Introduction 4.2. Relationships among objects in an inheritance hierarchy 4.3. Assigning reference of subclass to superclass-type variable 4.4. Assigning a superclass reference to subclass-type variable 4.5. Subclass method calls via superclass-type variable 4.6. Summary of allowed assignments between superclass and subclass variables 4.7. Multiple inheritance and interfaces	After completing this chapter, the students are expected to: ➢ Understand the concept of Polymorphism

Week 5	**6 Hours**	**Chapter 5: Exception Handling** 5.1. Exception handling overview 5.2. The causes of exceptions 5.3. The Throwable class hierarchy 5.4. Handling of an exception 5.5. The throw statement 5.6. The finally clause 5.7. User defined exceptions	After completing this chapter, the students are expected to: ➢ Understand the concept of Exception Handling
Week 6	**9 Hours**	**Chapter 6: Graphical User Interface (GUI)** 6.1. Introduction AWT Packages 6.2. Layout Managers 6.3. Event Handlers and Listeners 6.4. Java Swing APIs 6.5. Basic GUI Controls 6.6. Menus and Menu Bars 6.7. Summery which includes GUI Component classes, GUI Container classes and Custom graphics classes.	After completing this chapter, the students are expected to: ➢ Understand the concept of Graphical User Interface (GUI) ➢ Understand Event Handlers and Listeners
Week 7	**8 Hour**	**Chapter 7: File and Database management** 7.1 File handling 7.1.1. Introduction 7.1.2. I/O classes 7.1.3. File and File Dialog objects 7.1.4. Random Access files 7.2. Database management	After completing this chapter, the students are expected to: ➢ Understand the concept of File and Database management ➢ Understand File Handling

Teaching Strategy	The course will be delivered in the form of lectures, lab session, group discussions, and individual and group project works.
Role of Instructor(s)	Delivers lectures, conduct lab session, prepares reading assignments and topics for group discussion, prepares projects by discussion with student, gives consultation and advises students on project works and assignments, prepares and evaluates quiz, assignment, midterm and final examination.
Role of Students	Attend lectures, lab session and presentation, work in team on group work, participate in group discussion, discusses with the instructor on topics of interest for project work, delivers and presents project work, attend quiz, midterm and final examination.
Required software and/or hardware	Desktop computer, JDk(Java development kit) and Net Beans

Course Assessment Methods

Continuous Assessment Method		Assessment date	Chapters	Weight
Tests	Test – 1	Week -2	C1 and C2	10%
	Test - 2	Week-4	C3 and C4	15%
Assignment		Last Week		15%
Project		Last Week	All Chapters	20%
Final Examination		Based on Final exam schedule	All Chapters	40%
Total				100%

References

- ✓ Ethio open courseware https://eopcw.com/
- ✓ Java tutorial https://www.javatpoint.com/java-tutorial
- ✓ W3 school https://www.w3schools.com/java/default.asp
- ✓ Y. Daniel Liang, 2009. Int. to Java Programming Comphrensive version, Seventh Edition. Atlantic State University
- ✓ Deitel, 2010, Java for Programmers Second edition. Prentice Hall
- ✓ Deitel, 2004. JAVA how to program. 5th Ed. New Delhi: Prentice-Hall of India ✓ R. Sahoo& G. Sahoo, 2001. JAVA 2 The complete reference book; New Delhi: Prentice-Hall of India
- ✓ C. Thomas Wu, 1999. An introduction to Object-Oriented Programming with Java; McGraw Hill.
- ✓ John Lewis & William Loftus, 1998. JAVA Software Solutions:- Foundations of Program Design; Addison-Wesley.