University of Computer Studies, Yangon
B.C.Sc./B.C.Tech.

CS-202 (Principle of Computer Science II)
(Major – Core)
Course Description

Course code number	CS-202	**Course Title**	Principle of Computer Science II
Semester hours	4 hours	**No. of Credit Units**	3
Prerequisite	CS-201 (Principle of Computer Science I)	**Course Coordinator**	Dr. Thinn Lai Soe Lecturer

Course Objectives

- To provide background information about object-oriented programming concepts.
- To explain syntax and semantic about primitive data types, selection statements, repetition statements, programmer defined classes, exception handling and assertions, characters and Strings, arrays of primitive data types and objects and arrays sorting and searching.
- To provide knowledge about all types of file I/O, from a low-level byte I/O to a high-level object I/O.
- To know object oriented properties: inheritance and polymorphism.
- To develop GUI and event-driven java applications.
- To develop Java applications using the standard java packages and object oriented feature.

Learning Outcomes

Upon the successful completion of this course, Student will be able to:

- Learn how to use standard classes and learn to design their own classes with object oriented features.
- Learn how to use the file I/O techniques.
- Learn how to develop Java applications using the standard java packages and object oriented features.

Reference Materials

1. An introduction to object-oriented programming with Java (Fifth Edition), C. Thomas Wu
2. Java 8 Programming Black Book (2015 Edition), DT Editorial Services, Comprehensive Problem Solver.
3. Learn to Program Java (Second Edition) by Danny C. C. Poo and Chee Seong Tan.
4. Programming with java A Primer (Fourth Edition) by E. Balagurusamy.
5. Introduction to Java Programming (Third Edition) by Y. Daniel Liang.

Course Organization

Student participation in the course will involve the following activities:

1. Attending the lectures
2. Practical Test
3. Assignment/ Tutorial
4. Moodle Test / Quiz
5. Exam

Assessment plan for the course

Exam (Moodle)	50 %
Practical Test	10 %
Assignment/ Tutorial	10 %
Moodle Test / Quiz	20 %
Class participation	10 %

Tentative Lecture
45 periods for 15 weeks (50 minutes for 1 period)

No.	Chapter	Page	Period	Detail Lecture Plan
	Chapter 1: Introduction to Object-Oriented Programming and Software Development	**1 to 28**	**1**	**Lectures + Assignment+ Practical**
1.	Introduction to Java Classes and Objects Messages and Methods Class and Instance Data Values		1	Explain Features of Java Programming
	Chapter 2: Getting Started with Java	**30 to 83**	**2**	**Lectures + Assignment+ Practical**
2.	The First Java Program Program Components Edit-Compile-Run Cycle Sample Java Standard Classes Sample Development		2	Explain with example programs
	Chapter 3: Numerical Data	**85 to 149**	**3**	**Lectures + Assignment+ Practical**
3.	Variables Arithmetic Expressions Constants Displaying Numerical Values Getting Numerical Input The **Math** Class Random Number Generation		2	Explain with example programs
4.	The **GregorianCalendar** Class Sample Development		1	Explain with example programs
	Chapter 4: Defining Your Own Classes-Part 1	**151 to 119**	**2**	**Lectures + Assignment+ Practical**
5.	Defining and Using a Class Defining and Using Multiple Classes Matching Arguments and Parameters Passing Objects to a Method Constructors Information Hiding and Visibility Modifiers Class Constants and Local Variables Calling Methods of the Same Class Changing Any Class to a Main Class Sample Development		2	Explain with example programs
6.	**Practical Test**		**1**	Revised from chapter 1 to 4
	Chapter 5: Selection Statements	**221 to 301**	**2**	**Lectures + Assignment+ Practical**

7.	The **if** Statement and Nested **if** Statements Boolean Expressions and Variables Comparing Objects and The **switch** Statement Drawing Graphics and Enumerated Constants Sample Development		2	Explain with example programs
	Chapter 6: Repetition Statements	**303 to 371**	**4**	**Lectures + Assignment+ Practical**
8.	The **while** Statement Pitfalls in Writing Repetition Statements The **do–while** Statement		2	Detail Explain with example programs
	Loop-and-a-Half Repetition Control The **for** Statement Nested **for** Statements Formatting Output			
9.	Loan Tables Estimating the Execution Time Recursive Methods (*Optional*) Sample Development		2	Detail Explain with example programs
10.	**Practical Test**		**1**	Chapters 5 and 6
	Chapter 7: Defining Your Own Classes-Part2	**373 to 443**	**2**	**Lectures + Assignment+ Practical**
11.	Returning an Object from a Method The Reserved Word **this** Overloaded Methods and Constructors Class Variables and Methods Call-by-Value Parameter Passing Organizing Classes into a Package Using Javadoc Comments forClassDocumentation The Complete Fraction Class Sample Development		2	Detail Explain with example programs
	Chapter 8: Exceptions and Assertions	**445 to 493**	**4**	**Lectures + Assignment+ Practical**
12.	Catching Exceptions Throwing Exceptions and Multiple **catch** Blocks Propagating Exceptions Types of Exceptions		2	Detail Explain with example programs
13.	Programmer-Defined Exceptions Assertions Sample Development		2	
14.	**Practical Test**		**1**	Chapters 7 and 8
	Chapter 9: Characters and Strings	**495 to 553**	**4**	**Lectures + Assignment+ Practical**
15.	Characters and Strings Pattern Matching and Regular Expression The **Pattern and** Matcher Classes Comparing Strings		2	Detail Explain with example programs
16.	StringBuffer and StringBuilder String Processing and Bioinformatics Sample Development		2	
	Chapter 10: Arrays and Collections	**555 to 631**	**3**	**Lectures + Assignment+ Practical**

17.	Array Basics Arrays of Objects The For-Each Loop Passing Arrays to Methods Two-Dimensional Arrays		2	
18.	Lists and Maps Sample Development		1	Detail Explain with example programs
19.	**Practical Test**		**1**	Revised from chapter 9 to 10
	Chapter 12: File Input and Output	**685 to 731**	**3**	**Lectures + Assignment+ Practical**
20.	**File** and **JFileChooser**Objects Low-Level File I/O High-Level File I/O		2	Detail Explain with example programs
21.	Object I/O 709 Sample Development		1	Detail Explain with example programs
22**.**	**Practical Test**		**1**	Chapter 12
	Chapter 13: Inheritance and Polymorphism	**733 to785**	**3**	**Lectures + Assignment+ Practical**
23.	Defining Classes with Inheritance Using Classes Effectively with Polymorphism Inheritance and Member Accessibility Inheritance and Constructors		2	Detail Explain with example programs
24.	Abstract Superclasses and Abstract Methods Inheritance versus Interface Sample Development		1	Detail Explain with example programs
25.	**Practical Test**		**1**	Chapter 13
	Chapter 14: GUI and Event-Driven Programming	**787 to 879**	**7**	**Lectures + Assignment + Practical**
26.	Simple GUI I/O with JOptionPane Customizing Frame Windows GUI Programming Basics Text-Related GUI Components		2	Detail Explain with example programs
27.	Layout Managers Effective Use of Nested Panels Other GUI Components		2	Detail Explain with example programs
28.	Menus		2	
29.	Handling Mouse Events		1	
30.	**Revision**		**1**	**For the whole lecture**

Grading System

UCSY follows a letter grade system comprising of grades A, A-, B+, B, B-, C+, C, C-, D and F. All marks obtained by students during the semester will be used in the grading process. For undergraduate students, a grade of “C” or better is required in this course because it is a prerequisite for other courses in the program. **The student who gets the grade point less than 2 must take Re-Exam**.

The grading scale for this course is:

Marks obtained	**Letter Grade**	**Grade Point**
>=90	A	4
85-89	A-	3.75
80-84	B+	3.25

75-79	B	3
70-74	B-	2.75
65-69	C+	2.25
60-64	C	2
55-59	C-	1.75
50-54	D	1
0-49	F	0

Fail Grade and Re-Exam: C-, D, F (Grade point <2)

Class Attendance and Participation Policy:

- **Attendance**
  Class attendance is mandatory. Most of the material you will learn will be covered in the lecturers, so it is important that you not miss any of them. You are expected to show up on time for class, and stay for the whole lecture. Students are expected to attend each class, to complete any required preparatory work (including assigned reading) and to participate actively in lectures, discussions and exercises.
- Mobile phones must be silenced and put away for the entire lecture unless use is specified by the instructor. You may not make or receive calls on your cell phone, or send or receive text messages during lectures.
- You are responsible for all materials sent as email. Ignorance of such material is excuse. You are responsible for all materials presented in the lectures.
- Your conduct in class should be conducive towards a positive learning environment for your classmates as well as yourself.
- **Assignment, Quizzes, Moodle Test and Labs**
  We will take a short 3 to 5 quiz for every lecture and 20 points quiz moodle test after one or two chapters. Any assignment or quiz is simply missed, regardless of the reason why (e.g. illness, work, traffic, car trouble, computer problems, death, etc), and earns a grade of zero. You are strongly encouraged to complete all assignments and attend all quizzes so that you can check that you understand the material and can throw out bad grades, or grades for which you had to miss an assignment or quiz for a valid reason. Late submissions will not be accepted for any graded activity for any reason.
- There are **no extra credit opportunities**.
  Students may not do additional work nor resubmit any graded activity to raise a final grade.
- **Test**

  Test will start after two or three chapters finished and the coordinator will announce the date for the test.
- **Project**
  Project will start after finishing the whole course. You will have to make own project using the lecture. After finishing the project, you will have to make project presentation for about 15 minutes and there will be coding test for each person after presentation. You are responsible for your project about able to explain your system and your code, able to change your code at once when the teachers wish to amend your code.