

NOVA COLLEGE-WIDE COURSE CONTENT SUMMARY

ITP 220 – JAVA PROGRAMMING II (4 CR.)

Course Description

Imparts instruction in application of advanced object-oriented techniques to application development using Java. Emphasizes database connectivity, inner classes, collection classes, networking, and threads. Lecture 4 hours per week.

General Purpose

This course provides a comprehensive foundation sufficient for a student to create new and / or modify existing Java programs containing advanced Java classes and components.

Course Prerequisites/Corequisites

Prerequisite: ITP 120

Course Objectives

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

- Design, develop, code and test Java Programs using classes and techniques introduced
- Create classes and interfaces for use in solving problems
- Apply Java techniques and concepts introduced to solve problems

Major Topics to be Included

- Review of Java and object-oriented programming
- Creating Graphical User Interfaces (GUI's)
- Inner / outer classes and anonymous inner class event handling
- Abstract classes and Interfaces
- Polymorphism
- Regular Expressions and Java usage capabilities
- Exception handling
- Java input/output Text and Binary I/O
- Generics and Collection classes
- Threads
- Networking
- JDBC Database connectivity

Student Learning Outcomes

Review of Java and object-oriented programming

- Describe Java encapsulation, class creation and creating objects
- Write statements using class static variables and static methods
- Describe usage of the this keyword
- Explain the difference between primitive and reference data types
- Describe Java class inheritance
- Write statements using the super keyword

Advanced Graphical User Interface (GUI)

- Write code to create a GUI
- Explain the process to Layout components in a GUI
- Write code to create and add components to a GUI
- Write code to add event handling capability to GUI components and process events

Inner / Outer classes and anonymous inner class event handlers

- Explain the difference between inner and outer classes
- Write code to create and use anonymous inner class event handlers

Abstract classes and Interfaces

- Explain use of existing Java predefined abstract classes
- Write code using Java abstract classes and create abstract classes and methods
- Write code using predefined Java Interfaces and to create new Interfaces
- Write code using the Comparable Interface

Polymorphism

- Explain how polymorphism is used and write code using polymorphism

Regular Expressions

- Explain how regular expressions are used in Java
- Write code using regular expressions

Exception handling

- Write code to implement try catch and finally blocks
- Explain the difference between the throw and throws keywords
- Write code to catch, throw or re throw an exception
- Write code to create a custom Exception

Java input/output – Text and Binary I/O

- Explain the difference between Text and Binary files
- Explain use of the java.io APIs
- Write code to read and write both binary and text files
- Write code to use serialization and read and write objects
- Write code to read and write random access files

Generics and Collection Classes

- Explain how and why generics are used in Java
- Write code using the data structures provided in java.util collection classes
- Specifically Sets, Lists and Maps
- Relate collection classes with standard computer science data structures
- Write code using Iterators with collection class objects
- Write code using the Collections class methods to operate on collection class objects

Threads

- Explain the advantages of threaded programming
- Write Thread code by implementing the Runnable Interface and also by extending the Thread class
- Describe the potential problems of threading and to identify critical sections
- Write synchronized thread code

Networking

- Describe the java.net API
- Write code to implement a simple Multi-Threaded networking program

JDBC Database Connectivity

- Explain the use JDBC drivers
- Describe the program steps necessary to connect to a database
- Write code to read and write data to a relational database
- Describe the java.sql APIs
- Write code to execute all types of SQL statements from Java programs
- Write code to do prepared statements and stored procedures in Java

Required Time Allocation per Topic

In order to standardize the core topics of ITP 220 so that a course taught at one campus is equivalent to the same course taught at another campus, the following student contact hours per topic are required. Each syllabus should be created to adhere as closely as possible to these allocations. Of course, the topics cannot be followed sequentially. Many topics are taught best as an integrated whole, often revisiting the topic several times, each time at a higher level. There are normally 60 student-contact-hours per semester for a four credit course. (This includes 15 weeks of instruction and does not include the final exam week so $15 * 4 = 60$ hours. Sections of the course that are given in alternative formats from the standard 16 week section still meet for the same number of contact hours.) The final exam time is not included in the time table. The category, Other Optional Content, leaves ample time for an instructor to tailor the course to special needs or resources.

Topic	Hours	Percentage
Review of Java object-oriented programming	5	8.33%
Graphical User Interface programming and event handling	5	8.33%
Inner/outer classes and anonymous inner class event handling	3	5.00%
Abstract class and Interfaces and Polymorphism	5	8.33%
Exception handling	5	8.33%
Regular Expressions and Java usage capabilities	5	8.33%
JAVA input/output Text and Binary I/O	7	11.667%
Generics and Collection classes	5	8.33%
Threads	5	8.33%
Networking	5	8.33%
JDBC Database connectivity	5	8.33%
Other Optional Content	5	8.33%
Total	60	100%