# Macintosh HD:Users:deepa:Downloads:sps_long_color.jpg

# **Introduction to Java Programming**

# **General Course Information**

**Assistant Professor:** Sam Sultan

**Email:** sam.sultan@nyu.edu

**Course title/number:**  Introduction to Java Programming/INFO1-CE9238.001

**CEUs:** 3

**Semester/Year**: Fall 2019

**Class location:** TBD

**Class meeting dates/times:** Mondays, (Dates TBD)

6:00PM - 9:30PM

**Office Hours** By appointment

# **Course Description:**

Learn the fundamental concepts of object-oriented programming and the Java programming language. Understand Java’s write-once, run-anywhere philosophy and the Java Virtual Machine, which allows any Java program to run on any platform. Gain a solid understanding of the Java language syntax and semantics, including Java program structure, data types, program control flow, String and Arrays, defining classes and instantiating objects, information hiding and encapsulation, inheritance, method overloading and overriding, exception handling, input/output data streams, and UI components. Reinforce these new concepts with hands-on assignments and a complete Java programming project.

# **Course Prerequisites:**

None. (Knowledge of another programming language is a plus, but not necessary).

# **Minimal Tech Requirements:**

Student must be familiar/comfortable using a text editor (a text editor such as Notepad on the Windows platform or TextEdit on the Mac). Preferably, student should install a more robust text editor such as Atom or Sublime. Student must also be familiar/comfortable using the Windows DOS/CMD environment, or the Mac Terminal environment. A quick cheat sheet will be provided to facilitate the navigation and learning of some of those DOS/Terminal commands.

# **Course Structure/Method:**

This course will be delivered on Monday evenings in person. The class will encompass lectures, assignments, examples and demos, midterm and final exams. All class content and assignments will be made available online via the web at <http://oit2.sps.nyu.edu/~sultans/java>

# **Course Learning Outcomes:**

By the end of this course, students will be able to:

# Differentiate between Procedural vs. Object Oriented programming style.

# Understand and work with the various Java Data Types

# Obtain a solid understanding of how to control program flow

# Master the use of conditional processing, loops & iterations, and method creation and calls

# Gain knowledge of Objects and Classes

# Learn how abstraction, encapsulation, inheritance and polymorphism work

# Create method overriding and overloading to enhance your Object Oriented coding techniques

# Appreciate Java Exception handling paradigm

# Work with input and output files and streams

# Obtain an initial learning of Swing user interface and applets

# **Communication Policy:**

Credit students must use their NYU email to communicate. Non-degree students do not have NYU email addresses. NYU Classes course-mail supports student privacy and FERPA guidelines. All email inquiries will be responded to within 24 hours during Monday through Friday 5pm. Email sent on Saturday or Sunday will not be responded to until Monday. I will respond to you using NYU email.

# **Course Expectations:**

Students are expected to participate in each class session by offering their understanding of the subject, sharing ideas or discussing/commenting on another students comment. In addition, students must complete and submit all assigned homework on time. Late submission of homework will either not be accepted, or will result in a lower grade. Students are also expected to develop with and present a team project with other students, as well as take and pass a midterm exam and a final exam.

See full detail of expectations under “Assessment Strategy” below. Further information about specific assignments can also be found in the “Course Outline” section.

**Attendance:** Students are expected to attend all classes. Excused absences are granted in cases of documented serious illness, family emergency, religious observance, or civic obligation. In the case of religious observance or civic obligation, this should be reported in advance. Unexcused absences from sessions may have a negative impact on a student’s final grade. Students are responsible for assignments given during any absence. Each unexcused absence may result in a student’s grade being lowered by 1 point. A student who has three unexcused absences may earn a Fail grade

**Class Participation:** To receive full credit for the course, you should attend all classes since much of the learning occurs during class lecture, presentation and class discussions. Please contact the instructor if you anticipate missing any part of the class. Please arrive on time so as not to disturb the flow of the lecture. Excessive lateness may result in lower overall grade.

**Required & Recommended Material:**

* Beginning Java – 7th Edition
* **Authors** – Ivor Horton
* **Publisher** – Wrox Press Inc.
* Instructor will also provide session by session content available online at class website
* <http://oit2.scps.nyu.edu/~sultans/java>

**Additional Recommended Material**

* Just Java 2 – 6th Edition
* **Authors** – Peter Van Der Linden
* **Publisher** – Sun Microsystems Press.
* Thinking in Java – 4th Edition
* **Authors** – Bruce Eckel
* **Publisher** – Prentice Hall.

**Assessment Strategy:**

Contributing factors for determining your course grade include:

* Class Participation - **10%** *(Attendance is prerequisite to participation)*
* Attendance - **10%**
* Homeworks - **10%**
* Midterm Exam - **35%**
* Final Exam - **35%**
* Total 100%
* **Class Participation:** To receive full credit for the course, you should attend all classes since much of the learning occurs during class lecture, presentation and class discussions. Please contact the instructor if you anticipate missing any part of the class. Participation grades will be based on:
* Involvement in class discussions and activities
* Participation which demonstrates integration of reading, class work, relevance and application.
* Willingness to learn by accepting feedback, trying new skills and approaches, etc.
* Quality/quantity of providing effective and balanced class dialogue and feedback.
* **Homework:** Homework is always due one week from assignment date. Student should complete the assignment and bring their answers to class the following week. Instructor will ask students to present and discuss their answers prior to start of lecture of the following week.
* **Midterm Exam:** There will be a midterm exam. The exam will be an open book, open notes/internet style exam. The exam will test the student's acquisition of topics, concepts and competencies learned in this class up to mid-term.
* **Final Exam:** There will be a final exam. The exam will be an open book, open notes/internet style exam. The exam will test the student's acquisition of topics, concepts and competencies learned in this class. The final exam will only cover material covered in the second half of the term.

**NYU SPS Policies:**

“NYUSPS policies regarding the Family Educational Rights and Privacy Act (FERPA), Academic Integrity and Plagiarism, Students with Disabilities Statement, and Standards of Classroom Behavior among others can be found on the NYU Classes Academic Policies tab for all course sites as well as on the University and NYUSPS websites. Every student is responsible for reading, understanding, and complying with all of these policies.”

The full list of policies can be found at the web links below:

* University: <http://www.nyu.edu/about/policies-guidelines-compliance.html>
* NYUSPS: <http://sps.nyu.edu/academics/academic-policies-and-procedures.html>

**School Grading Policies:**

**NYUSPS Career Advancement (non-degree)**
<https://www.sps.nyu.edu/homepage/student-experience/policies-and-procedures.html>

###

### **Course Outline:**

**Session 1, Week 1, Introduction to Programming in Java**

* Introduction to Programming
* What is a Programming Language
* Compiled vs. Interpreted Languages
* Procedural vs. Object Oriented Languages
* The Java Environment
* The Java Compiler, The JVM Java Virtual Machine
* Downloading and Installing the SDK
* What is Path and Classpath
* Compiling and running a Java Program
* The Java Language Keywords

 **Reading:** Chapter 1

**Session 2, Week 2, Java Variable Data Types**

* Declaring Variables
* Java Primitive Data Types
* Variable Assignment
* Casting to another Data Type
* Automatic Data Type Promotion
* Mathematical Operators
* Operator Precedence
* Comments and Documentation

 **Reading:** Chapter 2

**Session 3, Week 3, Program Logic**

* Decision Making
* The if statement, and the switch statement
* Loops and Iterations
* The for and while statements
* Boolean Expressions
* Logical Operators
* Conditional Operators
* Variable Scope
* Creating and Using Methods

 **Reading:** Chapter 3

**Session 4, Week 4, Strings and Arrays**

* The String class
* String and Substring Operations
* Comparing Strings for equality and inequality
* Array of Strings
* StringBuffer Objects
* Declaring Arrays
* Array of Arrays (Multi-dimensional Arrays)
* Iterating through arrays
* Sorting arrays
* Sorting arrays with a comparator object

 **Reading:** Chapter 4

**Session 5-6, Week 5-6, Object Oriented Concepts and Programming**

* Object Oriented Concepts
* Defining Classes
* Defining Variables and Methods in a Class
* Instantiating Objects
* Defining Constructors
* Method Overloading
* Packages
* Access Control to Class Members
* Abstraction and Encapsulation

 **Reading:** Chapter 5

* **Midterm Exam** (at start of session 6)

**Session 7, Week 7, Extending Classes and Inheritance**

* Class Inheritance
* Abstract Classes
* Extending Classes
* What is a Superclass, what is a Subclass
* The Universal Superclass
* The toString( ) Method
* Determining the type of an Object
* Inheritance and Polymorphism

 **Reading:** Chapter 6

**Session 8, Week 8, Java Exceptions**

* What are Exceptions
* Type of Exceptions
* Handling Exceptions
* The try/catch Block
* Define and Throw your own Exception

 **Reading:** Chapter 7

**Session 9, Week 9, File Input and Output**

* Input and Output Streams
* Using Readers, using Writers
* Working with the File Object
* Testing for Files and Directories
* File Input and Output
* Dealing with Buffers
* Reading from Files, writing to Files

 **Reading:** Chapters 8-11

**Session 10, Week 10, Java UI, AWT and Swing Classes**

* **Final Exam**
* Creating windows
* Swing components
* Menus, text, buttons & other components
* Using containers
* The Flow Layout Manager
* Adding menus to windows
* Event Handling, Event Listeners
* Creating Applets
* Displaying Applets in web pages

 **Reading:** Chapters 17-19