**This word doc is dedicated to the understanding and mastery of Java programming.**

**And using Data Structures and algorithms.**

**Put in the work.**

**Chapter 01 – Introduction to computers, Programs and The Java programming language.**

* Computers do not understand normal human languages.

* A low level programming language can be referred to as an Assembly language.
* This low level assembly language than passes the code written to an assembler which then translates to machine language, also called binary.

**Understanding Java and the world wide web:**

* Many commercial uses need java.
* A lot of sites on the web who incorporate backend technology uses java.

* The java language specification provides the language and it’s semantics, for developing these programs.
* The API, also known as the library contains predefined java classes, and interfaces for developing Java applications.
* Java SE, Jave EE, Java ME
* Client side, server side, mobile.

**Beginning to actually write in Java…**

* A Java program will always be executed from the Main method’s class.

* The most common errors occur with Syntax errors.

**End of basic concepts of programming, follow source code written within Netbeans Project.**

**Directory named: “Data Struc Book” all source code created there.**

**Chapter\_02 : Elementary Programming:**

* Writing a program involves designing an algorithm. And then translating that algorithm into a instructions, or code.

* In relation to Java… think of the **task… as your class name. And then the class must have a main method** in order for your task to be accomplished inside.
* Think of calculating the area of a circle…
* **The radius times radius times pi… is equaled to the area.**
* The program must prompt the user to enter the radius two times and then it must take those values and write them into a variable.
* Lines within the program show you stages of your code.
* Some values for example of your variables may have different values assigned to them, referring to this may be called as “**tracing the program**”
* Reading the input from the user requires the use of the **Scanner class. System.out is used for outputs, and System.in is used for inputs. (import java.util.scanner)**

**Example for working with the scanner class…**

**Follow the: ElementryProgramming/InputtingArea package for the source code…**

**Multiple practise files have been created within the NetBeans project: Elementary programming.**

**Continuing programming: elementary programming concepts.**

* Chapter 2 contains more information about the data types within the Java language.

* Practise code **SalesTax program** will be continued to work with storing variables, using the Input object of the scanner class. Refer to the elementary programming package created within the netbeans environment.
* Basics of variables, working under classes, and the main method understood.
* Also, taking in inputs from the user is understood as well.
* The software development life cycle is a concept in which software solutions, products, etc are created by.

Find the requirement of your system, do an analysis of the system existing or the one you would like to create, do a design for the system, implement, test, deploy, maintain.

**COMPUTING LOAN PAYMENTS FOR SDLC APPROACH.**

**Moving into Generics, Stacks, etc for course understanding.**

* Using generics helps with detecting errors at runtime.

* Primitive types must not be used within these kinds of lists.
* Only reference types: instead of int – write integer.

**Lists, Stacks, Queues etc.**

* The collection interface, defines common operations for all of these mentioned above.

**What is a queue – follows the first in first out architecture.**

Follow the example code to demonstrate an understanding.