**Section 1: Define / Answer**

Pg. 217-224, Java Programming *A comprehensive Introduction*

[*http://www.javatpoint.com/method-overloading-in-java*](http://www.javatpoint.com/method-overloading-in-java)

[*http://beginnersbook.com/2013/05/method-overloading/*](http://beginnersbook.com/2013/05/method-overloading/)

*http://beginnersbook.com/2013/05/constructor-overloading/*

**Define-**

**Polymorphism-**

**Polymorphism- The word 'polymorphism' literally means 'a state of having many shapes' or 'the capacity to take on different forms'. When applied to object oriented programming languages like Java, it describes a language's ability to process objects of various types and classes through a single, uniform interface**

**Method Overloading-(Define and Give a short code example)**

**Method Overloading is a feature that allows a class to have two or more methods having same name, if their argument lists are different. In the last tutorial we discussed constructor overloading that allows a class to have more than one constructors having different argument lists.**

**Constructor chaining-(Define and Give a short code example)**

Calling another constructor in the same class from another constructor is called constructor chaining. By using this() we can call another constructor in the same class. Incase we want to call another constructor, this() should be the first line in the constructor. Below example shows code for constructor chaining.

/\* Constuctor overloading

StudentInfo(){

//give default

}

StudentInfo(int a, int b){

}

StudentInfo(String a, String b){

}

\*/

/\* Method overloading

void sum(int a, int b) {

}

void sum(int a, int b, int c) {

}

\*/

**ONLINE RESOURCE:** http://tutorials.jenkov.com/java-io/file.html

**Understand and define methods in the Java File class-**

java.io.File *file* = **new** java.io.File("*filelocation.filetype*");

System.out.println("Does it exist?" + file.exists());

System.out.println("The file has " + file.length() + " bytes ");

System.out.println("Can it be read?" + file.canRead());

System.out.println("Can it be written to?" + file.canWrite());

System.out.println("Is it a directory" + file.isDirectory());

System.out.println("Is it a file?" + file.isFile());

System.out.println("Is it absolute? " + file.isAbsolute());

System.out.println("Absolute path is " + file.getAbsolutePath();

System.out.println("Is it Hidden? " + file.isHidden());

System.out.println("Last Modified on " + file.lastModified());

System.out.println("Last Modified on " +

**new** java.util.Date(file.lastModified()));

Task 1-

**USE OBJECT ORIENTATED PROGRAM DESIGN TO SOLVE PROBLEM**

Create a class called StudentInfo.

The class should contain a constructor for student ID, First Name, Last Name, DOB, and address.

Create 10 student objects matching the criteria.

**Output:**

A Header Row

Then the 10 student objects in a .txt file

