Absolute Java Exam Cheat Sheet: Ch 6-13

**Try/Catch:**

* try{//code that throws errors}
* catch(Exception e){//code that runs if error}
* Use throws AnException for uncaught exception

**Throw:**

- Inside catch(){};

-if(condition equals true)

throw new NameOfException(“message”)

* can use e.getMessage() to call message.

**Finally:**

* finally{};
* Executes code inside no matter error or not.

**Writing your own exception:**

* public class NameException extends Exception{

public NameException(){

super(“message”);}}

**Create Arrays:**

int[][] arrayName = new int[rows][cols];

**Copy Constructor:**

public Person (Person other){

if (other != null)

name = other.name;}

**Sorting Arrays:**

* Ascending: Arrays.sort(array);
* Decensing: Arrays.sort(array, Collections.reverseOrder());

**Polymorphism:**

* Early binding: invoked during compiling.
* Late binding: use for all methods except private, final, and static methods during runtime.

**Clone Method:**

* Simple copy = (Simple).super.clone();
* copy.other = (person).other.clone();

**PrintWriter:**

* PrintWriter fout = null;

Fout = new PrintWriter(new FileOutputStream(filename)); (arg1 = filename)

* By default: overwrites file contents (arg2 = false)
* To append: Arg2 = true
* Use fout.close() or fout.flush() to flush to file

**I/O using Scanner:**

* Scanner fin = new Scanner(new FileInputStream(filename));
* Use nextInt() and nextLine()
* Use hasNextInt()
* Scanner fout = new Scanner(new FileOutputStream(filename));
* Use fout.close();

**I/O using BufferedReader:**

* BufferedReader fin = null;
* Fin = new BufferedReader(new FileReader(filename));
* Uses read() and readLine()

**Binary I/O:**

* ObjectOutputStream fout = new ObjectOutputStream(new FileOutputStream(filename));
* writeUTF(String aString) :writes a string
* writeChar(), writeDouble(), writeChar()
* ObjectInputStream fin = new ObjectInputStream(new FileInputStream(filename));
* readUTF(), readDouble(), readInt()

**Read/Write objects to a file:**

* SomeClass name = (SomeClass) objectInputStream.readObject();
* Or writeObject()
* Class of object being written must “implements Serializable”

RandomAccessFile (.bin)

-getFilePointer(), seek(), length(), close()

**Recursion:**

* if() //Stopping Case
* else call the method
* mystery(s.substring(0,s.length()-1));

**UML:**

* Unified Modeling Language
* (-)private, (+)public, (#)protected, (~)package
* Class Diagram:
  + Top: Contains Class Name
  + Middle: Data Specifications, variables
  + Bottom: Actions and Methods
* Does not define methods.

**Notes:**

* Inside a catch block, clear the buffer so that the error does not get thrown again.
  + KeyIn.nextLine();
* If an exception is thrown, message is still printed even without try or catch blocks.

**Split:**

* String[] parts = String.split(“delimeter”)