**Discussion Assignment Unit 01**

by [Victor Nwankpa](https://my.uopeople.edu/user/view.php?id=57369&course=3588) - Thursday, 9 April 2020, 7:54 AM

The Computer program which is otherwise known as Exception handler is responsible for handling exceptions in java.

When an exception occurs inside a Java method, the method creates an Exception object and passes the Exception object to the JVM (in Java term, the method "throw" an Exception). The Exception object contains the type of the exception, and the state of the program when the exception occurs. The JVM is responsible for finding an exception handler to process the Exception object. It searches backward through the call stack until it finds a matching exception handler for that particular class of Exception object (in Java term, it is called "catch" the Exception). If the JVM cannot find a matching exception handler in all the methods in the call stack, it terminates the program.

**Reference:**

Java Notes (n.d.) Java Programming Exception Handling & Assertion. Retrieved from: https://www3.ntu.edu.sg/home/ehchua/programming/java/J5a\_ExceptionAssert.html

### Discussion Assignment Unit 02

by [Victor Nwankpa](https://my.uopeople.edu/user/view.php?id=57369&course=3588) - Friday, 17 April 2020, 4:45 PM

**When should we consider using recursive algorithms when writing a program?**

**Discuss in terms of advantages and disadvantages**

**Advantages:**

1. Problems are often resolved in a straightforward way rather than a far larger redundancy solution.
2. Recursive algorithms is used to reduce unnecessary calling of function. This means that it is used instead of calling function severally through other methods.
3. The function does return for itself. Which calls itself back.

If the function calls itself severally or excessively it's going to cause a controversy within the memory of the device, because the function whenever it calls itself will occupy a particular amount of memory within the device, so if it calls itself too over and over (like a thousand times), the system memory could also be full and make the system too slow Until it completely stops working.

**Disadvantages:**

1. Usually not considered when the program is little and running on a PC.
2. Because the function has to add to the stack with each recursive call and keep the values there until the call is finished, the memory allocation is greater than that of an iterative function
3. The recursive solution is usually logical and it's very difficult to trace.

**Reference:**

**Eck, D. J. (2019). *Introduction to programming using Java, version 8.1*. Hobart and William Smith Colleges.**

**Using ADTs (abstract data types) has advantages for program modularity. Discuss**

**1) Abstraction:**

A very important method used to make it easy to write commands to programmers, it makes you able to do what you want without having to know all the details in which it was executed. Abstraction makes you deal with things superficially instead of delving into the details of complex codes.  It is a technique used to hide program execution details. To apply the concept of abstraction, we use the word abstract under specific conditions.

ADTs aid program modularity. ADTs operations are implemented once and if working as expected, all parts of the current program that need to perform ADTs operation will call on the services of the ADTs routine. The routine can even be transferred into a new project. Thus, fulfilling many of the objectives of program modularity.

**2) Localization of bugs:**

if there are bugs either in the representation or implementation of the ADT, the bugs are local to the ADT, and can’t be caused by code that uses the ADT.

**3) Localization of changes:**

if the programmer decides to change the representation of an ADT (i.e, to change the representation of a set from a linked list to a stack or array), then only the implementation of the ADT need be changed; code that uses the type is not affected.

**4) Encapsulation:**

encapsulation or “black boxing” ensures that data cannot be corrupted, and ensures that your code-base is free of errors

**5) Robustness:**

Your programs will be robust and have the ability to catch errors and prevent exceptions.

**References:**

**Eck, D. J. (2019). *Introduction to programming using Java, version 8.1*. Hobart and William Smith Colleges.**

**Medium (n.d.) Advantages of ADTs.  Retrieved from: https://medium.com/@chrisharwell72/advantages-of-adts-for-program-modularity-4449f9f08b07**

### Discussion Assignment Unit 04

by [Victor Nwankpa](https://my.uopeople.edu/user/view.php?id=57369&course=3588) - Saturday, 2 May 2020, 3:23 AM

**Specify advantages and disadvantages of using this method and suggest an alternative for equality.**

The java string equals() method compares the two given strings based on the content of the string. If any character is not matched, it returns false. If all characters are matched, it returns true.

**Advantages**

The advantage is that you can check two objects for equality i.e. if you need to see if two objects are the same.

**Disadvantages**

The disadvantage is that it can be very difficult to override correctly. I think that equality is even simpler. It is possible to determine the values to be equal, and if the result is equal, it is true, otherwise it returns false.

**An alternative for equality is** the use of the == operator to compare equality, but it may come with some unexpected results.

If you’re comparing Strings, you’re most likely wanting to know whether or not they contain the same sequence of letters and numbers. This is implemented in the .equals() method, which you could also override if you wanted to implement even more details.

The == operator would not look at the Strings content, but instead look at the pointer reference of the objects and determine whether or not they refer to the same place in memory. This means that you will only get the correct result if your variables are aliases.

**References:**

**Eck, D. J. (2019). *Introduction to programming using Java, version 8.1*. Hobart and William Smith Colleges.**

**Quora (May 8, 2018) A class object defines an equal() method.  Retrieved from https://www.quora.com/A-class-object-defines-an-equal-method-to-compare-objects-What-are-the-advantages-and-disadvantages-of-using-this-method-What-is-an-alternative-for-equality**

### Discussion Assignment Unit 05

by [Victor Nwankpa](https://my.uopeople.edu/user/view.php?id=57369&course=3588) - Thursday, 7 May 2020, 6:47 PM

**When writing a data structure what should be our guidelines for choosing the right Java Collection?**

Data structure is an organized collection of data, that can be treated as a unit in a program (Eck 2019, p.738). Java Collections are: Lists, Sets, Maps, Queue e.t.c.

1. choose the general type of organization that your data needs to have (e.g. map or list); without too much thought, this is usually fairly clear;

2. Then, choose the implementation of that type that has the minimum functionality that you actually require (e.g. don't choose a sorted structure if you don't actually need the data to be sorted).

3. While choosing a data structure in java, one should know how to use it and the complexity of the functions that are to be performed.

4. Choose such a data structure that enhances quality and readability of a code.

In general, the algorithm that underlies each collection class is designed to be a good tradeoff between efficiency and certain minimal requirements. So as long as you understand the minimal requirements that a given class is designed to provide, you shouldn't need to get too bogged down in the actual algorithms (though if you are interested in algorithms, the source code to all the collections classes is available and they make fascinating case studies, of course).

**References:**

**Eck, D. J. (2019). *Introduction to programming using Java, version 8.1*. Hobart and William Smith Colleges.**

**Javamex (n.d.) Collections, how to choose.  Retrieved from:**[**https://www.javamex.com/tutorials/collections/how\_to\_choose.shtml**](https://www.javamex.com/tutorials/collections/how_to_choose.shtml)

### Discussion Assignment Unit 06

by [Victor Nwankpa](https://my.uopeople.edu/user/view.php?id=57369&course=3588) - Thursday, 14 May 2020, 2:10 PM

**What are the differences of handling Socket and ServerSocket ?**

A socket is used by a program to establish a connection with another program on a network. Communication over a network involves two sockets, one on each of the computers involved in the communication. Java uses a class called java.net.Socket to represent sockets that are used for network communication (Eck, 2019, p. 575).

**Socket :**

1. Socket class is used for client side. we have to create an object of Socket class for any networking application

2. Socket class encapsulates the behavior of the active side. (i.e. the client)

3. This class implements client sockets. A socket is an endpoint for communication between two machines.

4. Socket class is available in java.net package

5. Server class, we can create like this:  Socket s = new Socket(ipaddress, portnumber);

**ServerSocket:**

1. ServerSocket class is used for server side. we have to create an object of ServerSocket class for any networking application to support server operations.

2. ServerSocket class encapsulates the behavior of the passive side (i.e the server)

3. This ServerSoket implements server sockets. A server socket waits for requests to come in over the network from clients.

4. ServerSocket class is available in java.net package

5. ServerSocket class, we can create like this:  ServerSocket server = new ServerSocket(portnumber);

**References:**

**Eck, D. J. (2019). *Introduction to programming using Java, version 8.1*. Hobart and William Smith Colleges.**

**Tutorials Point (n.d.) Java - Networking.  Retrieved from:**<https://www.tutorialspoint.com/java/java_networking.htm>

**The Model-View-Controller design pattern decouples data access, business logic, and data presentation and user interaction. Discuss.**

One of the principles of object-oriented design is division of responsibilities. Ideally, an object should have a single, clearly defined role, with a limited realm of responsibility. One application of this principle to the design of graphical user interfaces is the MVC pattern. “MVC” stands for “Model-View-Controller” and refers to three different realms of responsibility in the design of a graphical user interface (Eck, 2019, p. 691).

The MVC pattern decouples data access and business logic from data presentation and user interaction by introducing an intermediate component: **the controller**.

1.      **Model**- The model represents the information (the data) of the application. It can also represent the business rules used to manipulate the data.

The model's view pointer is weakly typed (only a base class pointer); the model should know nothing about the views which control it.

2.      **View** - The view corresponds to elements of the user interface, typically JSP pages and Struts Tiles.

The view knows exactly what kind of model it observes and it has a strongly typed pointer to the model, allowing it to call any of the model's methods.

The view may have a weakly-typed (base class) pointer to the controller. It should not call any methods in the controller other than those defined in the base class because you may want to switch out the controller for another.

3.      **Controller** - The controller manages details involving the communication to the model of user actions. In Struts, the controller is really the combination of a number of Actions and an Action servlet.

The controller has pointers to both the model and the view and knows the type of both. Since the controller defines the behavior of the triad, it must know the type of both the model and the view in order to translate user input into application response.

**References:**

Eck, D. J. (2019). Introduction to programming using Java, version 8.1. Hobart and William Smith Colleges.

Base 22 (n.d.) Model-View-Controller Pattern.  Retrieved from: <https://wiki.base22.com/btg/model-view-controller-pattern-3492.html>

### Re: Discussion Assignment Unit 08

by [Victor Nwankpa](https://my.uopeople.edu/user/view.php?id=57369&course=3588) - Thursday, 28 May 2020, 4:05 AM

**When developing a new software we should take into account how to create a specific "Look and Feel" per customer, What are the advantages of using Java GUI ?**

A GUI (Graphic User Interface) is a graphical representation in which the users can interact with software or devices through graphical icons. GUI is the most common type of user interface in use today.  The components include things like windows, scroll bars, menus, buttons, and icons.  Usually, a mouse is used to manipulate such components or, on “touchscreens,” (Eck, 2019, p. 13).

**Advantages of Graphic User Interface (GUI) are:**

1. In GUI more than one task can run simultaneously
2. It is a very user-friendly interface and easy to use
3. It is based on graphical representation e.g. images, and icons etc.  This is also known as “look and feel”
4. This interface made it possible for a mouse to be used.
5. Users do not need to know any programming languages like java, python etc in order to use a GUI.
6. It is also known as multi user operating system.
7. You do not have to learn complicated commands.
8. It is easy to explore and find your way around the system using a GUI interface

**References:**

Eck, D. J. (2019). Introduction to programming using Java, version 8.1. Hobart and William Smith Colleges.

Computer Ganga (n.d.) what are the advantages and disadvantages of graphical user interfaces.  Retrieved from: <https://computerganga.blogspot.com/2018/11/gui.html?m=0>