**Higher Diploma in Science in Data Analytics**

Multi Paradigm Programming - Shop Assignment

Comparing Programming Paradigms

*Roisin Fallon*



**Objective:**

The goal of this report is to compare the shop solution for two different programming languages C and Java. Within this report we will make sensible comparisons and discuss key differences and similarities between the two programming languages using shop example.

**Introduction:**

A programming paradigm refers to the way in which a problem is approached and will affect how the code is written and structured. In this project we will only be discussing procedural paradigm (in C) and the object orientated paradigm (in Java). All programming languages and paradigms are attempts to abstract low level details to allow programmes to solve problems at a higher or even different level. [3]

***Language Type***

The fundamental difference between C and Java is the use of different programming paradigm.

***C - Procedural programming language*.**

Procedural programming language will write procedures or methods to perform operations on the data.

Data used by a program can be put into local variable (inside a function) or global variable (outside a function). C is often considered a middle level programming language as it bridges the gap between assembly language and high-level languages. It differs from Java as it does not support inheritance which relates to the reusability of code. [1] C is designed to work cross platform. A standard compliant C program was developed with portability in mind and can be compiled for a variety of computer operating systems with few changes to its source code. However, it was not prefect and when Java was designed in 1991 it was designed in a way to provide strong cross platform support which they achieved.

***Java – Object Orientated programming language***

OOP refers to the language that uses objects in programming with the aim to bind together the data and the functions that operate in them so that no other parts of the code can access the data except the function. There are several key OOP concepts in Java which include the following:

1. Abstraction – hides underlying complexity of data. Reduces repetition.
2. Encapsulation – protect data stored in class from system wide access which is represented in the shop program by keeping the class variables private and providing public getter and setter methods to each of them.
3. Inheritance – a class (child) can extend another class (parent class) by inheriting its features. It reduces repetition and improves code reusability. [14]

The tagline for Java is “Write once, Run Anywhere”) the idea being that you could write your code once have it run on Java virtual machine and then that code could be taken and run on a Windows, Linux, Mac machine on any kind of operating system as long as that operating system supported the Java virtual machine. This has a distinct advantage over C.

Java is not considered a pure OOP language i.e. not everything is an object, mostly everything is an object except primitive data types which include int, char, float and double. What this means is we cannot primitives messages as we cannot invoke a method on primitives pass an integer. String in Java hides the complexity of using a character array to store a string of characters. It is slightly more high level and easier to work than it is in C. fields in that object).

**Format/Layout of files**

***Shop Example in C***

C follows a “top down” approach which places greater importance on the flow of the program instead of the data on which it operates.[1] The ideology behind the top down approach is to start with a big problem and break it down into sub-problems, these sub problems are broken down until the can be solved individually. Once solved they are then added together to get the complete solution to the problem. [5] This type of model has a high ratio of redundancy as the size of the project increases. In the shop example the more edits made to the code and the more lines of code added the harder it was to locate and edit all the related elements (essentially it is harder to maintain the code).

The shop program is presented in a single file for C (shop.c). Code could easily become untidy and difficult to maintain if the shop program was much bigger which is a disadvantage of C. As the shop program in procedural programming is coded in one file the order in which the commands are coded is important as they are executed starting from the top of the file to the bottom of the file.

Struct in C can provide a structure to store related pieces of information e.g. Product struct – name and price. However, it does not provide a way to store the functionality alongside the data. In other words, there is no way of putting methods or functions inside the struct and that is the primary difference between a struct and an object. A struct is just data and an object in Java is state and functionality. Note modifications to the state of a struct are done by methods outside of the struct.

Strings in C can be created easily however concatenating these strings is a little more difficult as the user must keep track of how much memory a string uses which is manually allocated by the user. Java on the other hand is much simpler as there is no need to worry about the size of the string and when there are no more references to the string the memory is reclaimed automatically by the garbage collector.

***Shop Example in Java***

Java follows a “bottom up” approach meaning it starts with basic functionality using existing primitives of the programming language and constructing gradually more complex features until the shop program has been fully coded. [12] The java shop example starts by designing the class from the basic level of the programming feature and then moving to the main part of the program.

The shop program in Java is designed so that each class has a separate file, that was named the same as the class itself. Initially this seemed more complicated when compared to C (single file) but with time I noticed it had several advantages. Java’s design made it easier to navigate and pull or update information relating to a class. Java has additionally functionality within the class through, getter and setter which are methods that are used to retrieve and update value of a variable. A setter is a method that updates value of variable and getter method reads the value of a variable. [6]

In the shop program getter and setters were used to update value of customer budget, shop float and product stock following a transaction. Less emphasis needed to be placed on the order in which methods within each class was coded in order to run the program which is another key difference to C.

OOP creates objects that contain both objects and methods. Data in stored in properties and logic is stored in methods. Properties and methods that work closely together live in the same class [13]. Classes and objects are the 2 main aspects of OOP. Objects are representation entities. An object has both state and functionality. The class of an object can be thought of as a blueprint or protype from which objects are created. It can both specify what state the object can have and what functionality the object can perform. In the shop example the class shop has shop cash and shop stock.

Constructor in Java are a special method that are used to initialise objects. It is called when an object of a class is created. It can be used to set initial values for object attributes. [9] This concept can be seen in the shop program where constructors were created in each of the classes. For example, in the Product class the constructor is taking in the name and price and storing them inside our object. Another example is in the Shop class where the constructor uses a string to take in the name of the file that we want to work with and will rad the initial values for the object from the file we selected.

**Compiled vs Interpreted**

Java is an interpreted language whole C is compiled. Compiled language infers that C is converted directly into machine code that will be executed by the processor. It has the advantage of executing code more efficiently usually and allows the user to have more control over memory management and CPU usage. Java runs through the shop program line by line and execute each command. As interpreters execute the source program code themselves, the code is platform independent (more portable). [11]

**Memory Management**

A pointer is essentially a simple integer variable which holds a **memory address** that points to a value, instead of holding the actual value itself. The computer's memory is a sequential store of data, and a pointer points to a specific part of the memory. [4]

***Shop Example in C:***

One of the initial problems I had with the shop program in C was when I passed the structs indirectly the changes were not maintained to shop and customer balance in C and stock balance following a transaction. When I changed something in a struct inside a method that change was not kept because what C does under the good is it passes in a copy of the struct so whatever changes get made to it get destroyed when the function finishes.

Pointers in C are used to maintain changes made after the method/function finishes. In the shop example a pointer to update the customer (budget) and shop (stock and float) to reflect the customer transaction was required. This was achieved in the processOrder function where pointers to the customer and shop are passed; processLiveOrder function where a pointer to the shop was passed.

***Shop Example in Java:***

Java has high memory and processing requirements. Thus, hardware cost is higher than in C. Java does not support the use of pointers. Java memory is managed internally by a compiler called the garbage collector. The main advantage this has over C is that it allows users to create new objects without explicitly allocating and deallocating memory by automatically reclaiming memory for reuse. [7]. This makes Java memory efficient and avoids memory leaks. [[8](https://www.edureka.co/blog/garbage-collection-in-java/)] Note Java uses references which is considered a special variable which points to an object. You can only reference objects for instance it is not possible to have a reference to an int.

**Exception Handling:**

One significant advantage Java has over C is how it deals with exceptions. Normal flow of program can be achieved in Java when an error arises through its powerful error handling mechanism. C does not have a built-in mechanism for dealing with exceptions.

This became apparent in the shop example. When an error occurred in Java the error was fixed much more easily and efficiently then in C as Java often provided detail about why the error happened which we could then research and rectify. In C a function wither returned some error code (when an error is expected) or the program crashed (usually long after the error occurred). [1]

**Conclusion**

Although I found the coding section in C time consuming, I could see that it made coding the Java section of the project much easier. C is a more basic and core computing language than Java. Initially I believed that C was an outdated programming language and that Java was superior in every way. However, I can see that C has its advantages and is involved in several important applications. C is used to develop MySQL databases (which I have personally used in other modules on this course), Google Chromium (developing browsers and extensions). It is also used in developing system and desktop applications and operating and embedded systems and is good for hardware dependent programming. [10] Java language works at a higher level than C and allows users to form standard programs and reusable code. It has a multi thread environment which allows users to perform multiple tasks at the same time in the program. It too has several important applications some of which include android apps, server-side technologies e.g. Apache and Big Data Analytics [10]. Below I have included a summarised table of the major differences between C and Java which I have explained in detail earlier in the report:

|  |  |  |
| --- | --- | --- |
|  | **C** | **JAVA** |
| **Language Type** | Procedural | Object Orientated |
| **Translator** | Compiled | Interpreted |
| **Memory Management** | No memory management | Memory management |
| **Memory Address** | Pointers | References |
| **Exception Handling** | Error Codes | Exceptions |

# **Bibliography**

1. <https://www.educba.com/c-vs-java/>
2. <https://www.edureka.co/blog/difference-between-c-c-and-java>
3. <https://learnonline.gmit.ie/pluginfile.php/98423/mod_resource/content/0/MPP%20-%202%20-%20What%20is%20a%20Programming%20Paradigm%3F.pdf>
4. <https://www.learn-c.org/en/Pointers>
5. <https://www.javaassignmenthelp.com/blog/java-vs-c/>
6. <https://www.codejava.net/coding/java-getter-and-setter-tutorial-from-basics-to-best-practices>
7. <https://www.dynatrace.com/resources/ebooks/javabook/how-garbage-collection-works/>
8. <https://www.edureka.co/blog/garbage-collection-in-java/>
9. <https://www.w3schools.com/java/java_constructors.asp>.
10. <https://www.guru99.com/difference-between-java-and-c.html>
11. <https://guide.freecodecamp.org/computer-science/compiled-versus-interpreted-languages/>
12. <https://stackoverflow.com/questions/881064/top-down-and-bottom-up-programming>
13. <https://academind.com/learn/javascript/functional-vs-oop-vs-procedural/>
14. <https://raygun.com/blog/oop-concepts-java/#inheritance>