

Shop Assessment Multit- Paradigm Programming

Java vs. C programming comparisons

52960 -- MULTI-PARADIGM PROGRAMMING

G00302722
Aidan Summerville

Table of Contents

Introduction	3
Programming paradigms.....	3
Procedural programming	3
Object oriented programming	4
Code	4
Similarities.....	5
Statement Terminators.....	5
Comments.....	5
Assignment.....	6
Conditional Statements	6
For loops	7
Other Similarities	7
Differences.....	7
Pointers	7
Memory allocation.....	8
Repositories	8
Input & Output.....	8
Other differences	9
Conclusions	10
References	10

Shop Assessment Mult-Paradigm Programming

Introduction

The brief for this project was to create a program that replicates an automated transaction between a shop and a user. The program must take an input shopping list in the form of a .csv file and compare the input shopping list to the shops stock, which is also stored in a .csv file. The program should give appropriate errors eg (item not in stock, user does not have enough cash, you have order to much of one item). The shop should update the cash and current stock of the shop in the stock.csv after each transaction. There must also be a life input method where the user can input the item they would like from the terminal.

The project brief requires that the shop programme be built in two different programming paradigms to establish which paradigm is better suited for this programming project. The two paradigms that will be used are, procedural programming using the C language and object oriented programming using Java. The two different instances should work in the same way and should be as similar as possible to the user in operation.

In the below report I will compare and contrast the differences and similarities of the two programming paradigms in relation to the implementation of this project. After the comparisons I will determine which language and programming paradigm is best suited for a program of this sort.

Programming paradigms

Programming paradigms are methods in which programs go about solving a problem or doing some task. Different programming languages are classed into programming paradigms by their features and the way in which these languages implement those feature to complete a task. Programming languages of the same paradigm should implement the same application using the same rules as each other and similar project by two languages of the same paradigm should look more similar than one of a different paradigm. There are a multitude of programming paradigms and some languages can be classified into more than one paradigm. Some programming paradigms and there language examples are:

- Procedural programming: BASIC, C, Pascal.
- Object-oriented: Java, Python, C++, Ruby.
- Declarative: SQL, XQuery.

Having knowledge of multiple programming paradigms allows a programmer the ability to decide which programming paradigm is best used with certain projects. In this project I used the procedural paradigm with C and object oriented paradigm with Java to determine which paradigm is best suited for a shop program.

Procedural programming

Procedural programming is a language that has step by step instructions that the computer follows in order to complete a task. This paradigm uses the top down approach (top down starts with your

main function and you think of the steps you need to take to divide up the code into smaller easier actions to complete the programs task). Some of the main features of procedural programming are predefined functions, local variables, global variables, modularity and parameter passing.

Advantages:

- The code can be reused in different parts of the program.
- The simplicity of the code and ease of compiling means that the code often runs much faster than other paradigms.
- Often uses much less memory.

Disadvantages:

- The code is much harder to write and can take much longer for a programmer to write a program using this paradigm.
- Syntax is often much further away from human readable than other languages.
- Data is exposed to the whole program making it unusable when security is a factor.

Object oriented programming

Object oriented programming has a collection of classes and projects which are use for communication in order to complete a given task. The importance is on data and not on the procedure in this paradigm. Languages that use this paradigm often use the bottom up approach (The bottom up approach pieces together smaller pieces of code to join them together to accomplish more complex problems).Some of the main features of procedural programming are encapsulation, abstraction, inheritance and polymorphism.

Advantages:

- Inheritance.
- Data security.
- Code reusability.

Disadvantages:

- Can be inefficient.
- Steep learning curve.
- Large program sizes.

Code

Below are examples of the differences and similarities between C and Java with code examples from the shop project.

Similarities

Statement Terminators

Both C and Java are semicolon terminated languages. They use semicolons to distinguish the end of a statement.

Below are some examples in C and Java from the shop project.

Both examples perform the same operation in each program, to print the initial menu list to the screen. Each line is semicolon terminated at the end of each statement in both languages. Allowing that statement to perform its operation and a new statement to begin on the next line.

C

```
// Prints out list of options
printf("CHOOSE A NUMBER:\n");
printf("-----\n");
printf("[1] Use ShoppingList CSV\n");
printf("[2] Live Mode\n");
printf("[X] Quit\n");
```

Java

```
// prints out list of options
System.out.println("Choose a Number");
System.out.println("-----");
System.out.println("[1] Use ShoppingList CSV");
System.out.println("[2] Live Mode");
System.out.println("[X] Quit");
```

Comments

Both C and java use // before single line comments and /* at the beginning of multiline comments and */ at the end of multiline comments.

Below are examples of single and multi-line comments from both the C and Java shop examples found in my code.

C

Single line comment

```
//updates stock csv
void update2(char *pN, double updateCash, int *var1, int var2)
{
```

Multi-line comment

```
/* function that compares input shoppingList verses product stock
 *Takes input pointers from Customer and Shop
 */
void printShopList(struct Customer *c ,struct Shop *s)
{
```

Java

Single line comment

```
// opens the temp.csv file to write to
FileWriter fw = new FileWriter(tempFile, true);
```

Multi-line comment

```
/* updates the stock csv with new quantity's
 * function takes a string for the file location the stock csv, takes the old
 * stock cash, the new stock cash , and an arraylist of product names and an
 * arraylist of new product stock quantity's
 */
public void update(String filepath, String oldCash, String updateCash, ArrayList<String>
name,
    ArrayList<String> newQ) {
```

Assignment

Both C and java use the equals symbol = for assignment within their respective languages.

Both of the below examples from the shop project perform the same task. Declaring product price by taking the inputted product name. Both languages use the equal's symbol for assignment of the price variable

C

```
//gets product price
double price = findProductPrice(s, c->shoppingList[i].product.name);
```

Java

```
// gets product price
double price = findPrice(p.getName());
```

Conditional Statements

Conditional statements in both C and Java are similar. Both if statements below perform the same task in C and java. An if followed by a condition in brackets followed by curly braces having some operation or operations to perform with the curly braces. The only difference between the two statements is where the two languages are getting their product stock quantities from.

C

```
// if the quantity input is greater than or equal to the number of this item in stock con
tinue
    if(q <= s->stock[j].quantity)
    {
```

Java

```
// if the quantity they want is less than or = to what we have in stock continue
if (amount <= productStock1.getQuantity()) {
```

For loops

For loops in both languages work in the same way. Both languages take an initialization (int i = 0);(int j = 0); , A condition for which to continue the statement,(i < name.size()); (j < s->index;) and an increment or decrement statement. Followed by curly braces with operations to be performed if the loop condition is met

C

```
//for loop that iterates through all of the updates stock names
for(int j = 0; j < s->index; j++)
{
```

Java

```
// for loop that loops through the csv going through each product name in it
for (int i = 0; i < name.size(); i++) {
```

Other Similarities

There are many syntax similarities between the two languages. This is to be expected as both languages are of the imperative format and java is from the C tree of languages and many of the features seen in C are seen in Java because of this. Data types, keywords, constants and variables all appear in both languages but they are implemented differently between the two.

Differences

Pointers

C uses pointers Java does not. Pointers are variables that are used to point to the location or address of another variable within the program. Pointers must be declared with *variable name. In java no pointer handling is required. -> must be used when referencing pointers in c. Java uses a (.) when referencing locations of variables.

C

```
void printShopList(struct Customer *c ,struct Shop *s)
```

```
//if the product name appears in both stock and shopping list
if(strcmp(c->shoppingList[i].product.name,s->stock[j].product.name)==0)
```

Java

Same operations in java.

```
public void processOrder(Customer c)
```

```
if (productStock.getQuantity() - productStock1.getQuantity() <= 0) {
```

Memory allocation

One major difference between C and Java is that in C memory allocation must be manually assigned. Java internally manages memory so the user does not have to. Having to predefine memory allocation means that sizes of the memory needed must be known before hand. This is particularly difficult in terms of a shop program where the size of the user's shopping list must be limited to a predefined number and the shops stock must do the same. This means this language doesn't bode well for the scalability of the program or the expansion of the shop to more and more products.

C

```
//declares memory allocation for product name
char *name = malloc(sizeof(char) * 100);
```

Java

This is done in the background of the java code so there is no example.

Repositories

Java features a large amount of repositories that can be used to create programs more easily. These repositories add a lot more functionality to the code and add to ease of use. C has limited added functionality and needs to hard code more often to achieve the same result.

C

```
#include <stdio.h>
#include<string.h>
#include <stdlib.h>
#include <stdbool.h>
#include <unistd.h>
```

Java

```
import java.io.*;

import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Scanner;
```

Input & Output

The input and output functions operate differently between Java and C. In Java scan texts in the next inputted character until enter is pressed. C takes the first word or number unless otherwise stated.

While outputting Java automatically goes to a newline after each statement. In c /n newline characters must be called each time.

C

Input

Declares variable list. Takes the next line in the terminal and saves it in list.


```
char list;  
scanf("%s",&list);
```

Must declare `^[^\\n]*` to take in spaces in strings otherwise it only takes the first word. It also read the next line in the terminal that hasn't been scanned so if there is a `printf` statement there it takes that in. There is a need for 2 `^[^\\n]*` in this case.

```
printf("Please enter a product name?\\n");  
//declares char pName for the input of the product name for the live mode  
  
char space;  
//scans the white space  
scanf("^[^\\n]*",&space);  
//scans the input  
scanf(" %^[^\\n]*",&pName);
```

Output

Prints text to the terminal. `/n` moves to a new line. New line characters must be expressed in C while there are automatically done at the end of each statement in java.

```
printf("Please enter the filepath to your Shopping List CSV\\n");
```

Java

Input

Opens a scanner to allow for user input

```
// create the scanner to take in user input  
Scanner scan = new Scanner(System.in);
```

Takes the next input and puts it into the string choice

```
// scans the inputted option into the terminal  
String choice = scan.next();
```

Takes in all characters until enter key is pressed

```
// ask the user for what they want to buy and save as string  
System.out.println("What product do you want to buy?");  
String productName = scan.nextLine();
```

Output

Prints to the screen and moves to the next line.

```
System.out.println("Please enter the filepath to your Shopping List CSV");
```

Other differences

Other differences include C being a low to mid level language and java being a high level language, C uses top down and Java uses bottom up programming approaches. Java is interpreted while C is compiled, Java supports exception handling C does not.

Conclusions

In conclusion while both programs run in a similar way and provide many similar functions, from creating both programs and comparing and contrasting their differences and similarities, Java is the best language is the better language to build the shop in and by because of this object oriented programming is the better paradigm for this project. There are a number of reasons I believe this to be the case, the first being that because memory has to be allocated manually in C it would make it unsuitable if the shop was to scale the size of its stock list which is currently limited to 10 items. If the stock size exceeded 10, memory allocation would have to be manually allocated within the code. The stock csv could be added to and no change would be necessary with the java code.

Another reasoning that Java would be a better option for this project is security. All the programs data is on one file in C. There are private classes in different files that are inaccessible in Java. This is particularly important when dealing with transactions and personal data. Other security risks in c are pointers and the non use of exception handling.

Another advantage of Java is its portability its take line of write once run anywhere would me accessibility to the shop of different systems more applicable.

C has the advantage of speed and file size. Neither off these is particularly important in this context especially if you need to give up security and scalability to use them. For these reasons I believe java to be overall the better choice for the shop project and by that decision the object oriented programming paradigm.

References

Anon, (2019). [online] Available at: <https://www.quora.com/Why-do-we-say-object-oriented-programming-is-a-bottom-up-approach> [Accessed 17 Nov. 2019].

Anon, (2019). [online] Available at: <https://www.quora.com/Are-there-any-similarities-between-C-and-Java> [Accessed 17 Nov. 2019].

Anon, (2019). [online] Available at: <https://www.quora.com/Which-language-is-considered-to-be-more-secure-C-or-Java> [Accessed 17 Nov. 2019].

Anon, (2019). [online] Available at: <https://www.quora.com/What-is-difference-between-Java-and-C> [Accessed 17 Nov. 2019].

EDUCBA. (2019). *C vs Java - 13 Most Important Comparison You Should Know*. [online] Available at: <https://www.educba.com/c-vs-java/> [Accessed 17 Nov. 2019].

En.wikipedia.org. (2019). *C (programming language)*. [online] Available at: [https://en.wikipedia.org/wiki/C_\(programming_language\)#%22Hello,_world%22_example](https://en.wikipedia.org/wiki/C_(programming_language)#%22Hello,_world%22_example) [Accessed 17 Nov. 2019].

En.wikipedia.org. (2019). *Declarative programming*. [online] Available at: https://en.wikipedia.org/wiki/Declarative_programming [Accessed 17 Nov. 2019].

En.wikipedia.org. (2019). *Java (programming language)*. [online] Available at: [https://en.wikipedia.org/wiki/Java_\(programming_language\)#Hello_world_example](https://en.wikipedia.org/wiki/Java_(programming_language)#Hello_world_example) [Accessed 17 Nov. 2019].

En.wikipedia.org. (2019). *Object-oriented programming*. [online] Available at: https://en.wikipedia.org/wiki/Object-oriented_programming [Accessed 17 Nov. 2019].

GeeksforGeeks. (2019). *Introduction of Programming Paradigms - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/introduction-of-programming-paradigms/> [Accessed 17 Nov. 2019].

Hackr.io. (2019). *What is Procedural Programming? Key Features of Procedural Programming*. [online] Available at: <https://hackr.io/blog/procedural-programming> [Accessed 17 Nov. 2019].

Investopedia. (2019). *Top-Down vs. Bottom-Up: What's the Difference?*. [online] Available at: <https://www.investopedia.com/articles/investing/030116/topdown-vs-bottomup.asp> [Accessed 17 Nov. 2019].

programming, T. and Martin, C. (2019). *Top down and Bottom up programming*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/881064/top-down-and-bottom-up-programming> [Accessed 17 Nov. 2019].

Resources.saylor.org. (2019). [online] Available at: <https://resources.saylor.org/wwwresources/archived/site/wp-content/uploads/2013/02/CS101-2.1.2-AdvantagesDisadvantagesOfOOP-FINAL.pdf> [Accessed 17 Nov. 2019].

SearchAppArchitecture. (2019). *What is object-oriented programming (OOP)? - Definition from WhatIs.com*. [online] Available at: <https://searchapparchitecture.techtarget.com/definition/object-oriented-programming-OOP> [Accessed 17 Nov. 2019].

Wikipedia Contributors (2019). *C (programming language)*. [online] Wikipedia. Available at: [https://en.wikipedia.org/wiki/C_\(programming_language\)](https://en.wikipedia.org/wiki/C_(programming_language)).

Wikipedia Contributors (2018). *Comparison of programming languages*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Comparison_of_programming_languages [Accessed 9 Dec. 2018].