SOMANATHAN SUBRAMANIYAN

STUDENT ID: G00364742

Multi-Paradigm Programming

Shop Assignment

2019

Table of Contents

[Q1: Comparisons 1](#_Toc24889182)

[Q1.1. C Programming – Procedural Programming Language – Key points 1](#_Toc24889183)

[Q2: Differences and Similarities 1](#_Toc24889184)

[Q2.1. Differences : 1](#_Toc24889185)

# Q1: Comparisons

## Q1.1. C Programming – Procedural Programming Language – Key points

* Middle-level language. This falls between machine level language and high-level languages.
* Compiled language. The code is converted into machine level language so that it could be understood by the system
* Normally breaks down into the functions
* Used for Systems and Application programming
* This language Supports pointers. Call by value and Call by reference is supported.
* The code is not usually portable across different Operating systems. It is usually platform dependent.
* Over-loading functionality is not supported
* Does not support Object Oriented Concepts
* Used in Embedded programming

# Q2: Differences and Similarities

## Q2.1. Differences :

* Java - There are number of tools available which provides rich functionality to support the application development like Eclipse and not many tools available in-case of C with similar functionalities
* Java - Number of rich in-built functionality and extended types available compared to C:

Key Observation from Shop Assignment

* + Handlining for “String” and the “Arraylist” are better in Java.
  + Scanning “string” inputs are made easy compared to C. In C, it requires multiple level of precautions like flushing already existing values and trimming the white spaces etc
  + The unique features like “inheritance” and “Class/Objects” reduced the number of lines of code
  + Error handling functionality is easy in Java
  + “Arraylist” is automatically manages the list without the need using additional index management.
* C has pointers and Java doesn’t.
* Other differences

|  |  |  |
| --- | --- | --- |
| Features | C Language | Java Language |
| Security | Limited | Built in the language |
| Memory address | Pointers | Reference |
| Data Structure | struct | Class |
| Allocating memory | Malloc | new |
| De-allocating memory | Free | Automatic garbage collection |
| Data hiding | Pointers and static | private |
| Graphics | Use external libraries | In-build library |