

Analyses of the C and Java

Understand the programming languages features are important than coding with these languages. Therefore in this essay I will try to compare C and Java by using some specific approach that are belong to context of the fundamental of the programming languages basic. C is procedural language. Means of the procedural is it organized as a sequence of procedure for manipulating data. This language take an input, manipulate it then give an output. Java is language which use object – oriented principles. Object oriented focus on the objects. Objects represent a real world entities or abstract concept. Object contain data and methods (function) together. There are 3 different approach in this essay for analyse and compare the Java and C languages. These are syntax, semantics and other features. For syntax I will compare tokens, punctuations, etc.. of these languages. At semantics part I will mention of types, control, loops, etc. In last and the most important part of the this essay is will be readability, writability, reliability, etc...

1. Syntax Analyses

1.1 Tokens

Tokens are the smallest unit of a programming language. Keywords, Identifiers, Literals, Operators, Punctuation and Comments are tokens of the programming languages. With these tokens programmer can design and implement code blocks. For instance ; “int x = 5 ;” block has 5 tokens. These are “keyword identifier operations literal punctuation” .

1.1.1 Keywords

C is low – level language and it is placed first programming languages. Therefore it doesn't contain some simplified keywords. At the below there are instance of the tokens and comparisons of the these 2 languages

C tokens:

```
#include<stdio.h>
typedef struct a
{
    int a;
}a;

class a
{
}

int main()
{
    int x=5;
    double a=1.1;
    char b='e';
}
```

C has all data types which are in the java. Additionally it has structure that use for collect different data types but it doesn't have class, because C is procedural language. Therefore programmer can't use principles of the object – oriented.

Java tokens:

```
public class j2 {

    typedef struct a
    {
        int a;
    }a;

    /**
     * Innerj2
     */
    public class Innerj2 {

    }

    Run | Debug
    public static void main(String[] args) {

        int x=5;
        double a=1.1;
        char b='e';
        boolean c=true;

    }

}
```

Java has same data type with C language. But Java is object oriented language. Therefore it has additional keywords such as public class. But it doesn't contain struct. Because it has already contain class.

1.1.2 Punctuation

Punctuation of the Java same as C language. Both of them use “{“, “}” for open and close code blocks or conditions or loops. And they use “;” for finish the lines

| | |
|---|---|
| <pre>int x=5; for(int i=0;i<x;i++) { System.out.println(x); }</pre> | <pre>int x=5; for(int i=0;i<x;i++) { printf("x"); }</pre> |
|---|---|

The picture at the left side is instance of the java and other one belong the C language.

We can observe the both language’s punctuations ways are very familiar.

1.1.3 Identifiers,Literals,Operators,comments

These tokens are completely same both of them . they use same principles while compile the tokens.

| | |
|---|---|
| <pre>Run Debug public static void main(String[] args) { int sample=5; int sample2=sample+5; }</pre> | <pre>int main() { int sample=5; int sample2=sample+5; }</pre> |
|---|---|

These to pictures are just instance . when we check the different code block for these tokens type, we obtain same result.

At below picture is instance of the comment

| | |
|---|---|
| <pre>//public static void main(String[] args) /*{ int sample=5; int sample2=sample+5; }*/</pre> | <pre>11 12 //int main() 13 /*{ 14 int sample=5; 15 int sample2=sample+5; 16 }*/</pre> |
|---|---|

1.2 Punctuations

In context of the language and grammar, “punctuations” refers to marks and symbols. These marks and symbols are use for indicate code structure first of all.it links structure with each other or it help programmer to complete the structure. Another task of the these punctuations is increase readability. Readability is the most important part of a programming language. Therefore we can

assume punctuations is as important as readability. Java and C languages has similiar punctuations. These languages use “;”, “,”, “.”, “()”, “[]”, “{}”, “< >”, “*” etc.. to apply punctuations . Symbols of the punctuations in Java and C almost same . There are little differents in thier punctuations symbols.

Java is a programming language that use object oriented principles. Therefore language forced programmer to create and use classes . because of that programmer obey the encapsulation principles automatically.

```
// JAVA
public class j2 {
    Run | Debug
    public static void main(String[] args)
    {
        int i=0;
        int j=0;

        for(;i<5;i++)
        {
            for(j=0;j<5;j++)
            {
                if(i==j)
                {
                    System.out.println(i);
                }
            }
            System.out.println();
        }
    }
}

8 // C
9 #include<stdio.h>
10
11 int main()
12 {
13     int i=0;
14     int j=0;
15
16     for(;i<5;i++)
17     {
18         for(j=0;j<5;j++)
19         {
20             if(i==j)
21             {
22                 printf("%d",i);
23             }
24         }
25         printf("\n");
26     }
27 }
```

It is obvious to these two languages almost has same punctuations symbols and structures. although both of them has similiar punctuations structure; java has additional structures. We committed the C is procedural language and java is object – oriented language. Picture at the above show us , java forced programmer to form class and apply principles of the object oriented .As a result java punctuations can create encapsulation . But C is programming language which use procedural rules . Therefore it can not crate encapsulation. On the other hand if programmer dint obey the programming rules , program has been written dont be readable. Beacuse of this issue only programmer can understand the program and in fact maybe he / she can not solve the issue in the program.

```
// JAVA
public class j2 {
    public static void main(String[] args)
    {
        int i=0;
        int j=0;

        for(;i<5;i++){
            for(j=0;j<5;j++){
                if(i==j){
                    System.out.println(i);
                }
            }
            System.out.println();
        }
    }
}

8 // C
9 #include<stdio.h>
10
11 int main()
12 {
13     int i=0;
14     int j=0;
15
16     for(;i<5;i++){
17         for(j=0;j<5;j++){
18             if(i==j){
19                 printf("%d",i);
20             }
21         }
22         printf("\n");
23     }
24 }

8 #PYTHON
9 for i in range(5):
10     for j in range(5):
11         if i == j :
12             print(i)
```

Picture at the above show us if programmer dont create clean structure or language dont force user to create clean structure . issues may arise. But python is good instance for this issue. Python forced user to use tabs for each structure. In this way a cleaner language emerges.

1.3 Data Types

Java and has same primitive types . actually all imperative languages has same primitive types. Beacuse primitive data types are basics of the programming languages. Although primitive types are basics of the programming, manipulaiton and usage of these data types can be more hard. Because of that some programming languages improve data structures to facilitate programming. Java is one of these languages. It has data sturctures to give programmer some programmin skills for facilitate programming and manipulatind data types. these are linked list , arraylist, queuse, stack etc.. thanks to object orianted principles java can provide these structures

1.4 Pointers

C is celebrated for its use of pointers, which allow user to acces memory and manipulate the content of memory adress. Although pointer give power to programmer , it gives additional syntax features.

```
// C
#include<stdio.h>
#include<stdlib.h>
int try(int *p)
{
    *p=54;
    return *p;
}
int main()
{
    int i=0;
    int *p=&i;

    int *p2=(int *)malloc(4*sizeof(int));
    int *p3=(int *)calloc(4,sizeof(int));

    int a=try(&i);
    int b=try(p2);

    free(p2);
    free(p3);
}
```

In this 20 lines. There are too much syntax and semantic knowledge about pointer. The truth is pointer is power. Programmer can access and manipulate every data with pointer. But if programmer doesn't have enough knowledge about pointer, he/she shoots himself in the foot. On the contrary the C language Java doesn't allow user to access everything. Instead of user Java performs some operation. Thanks to this Java program can be more safe and programmer can code easily.

2. Semantic Analyses

2.1 Memory Management

C is celebrated for its use of pointers, which allow user to access memory and manipulate the content of memory address. This feature can give programmer power, but it gives security problems and reliability issues with power. Programmer can allocate memory address or free memory with pointer. Also this feature creates some issues. Because of these weaknesses Java doesn't allow programmer to access memory directly or manipulate these memory addresses. Java controls all memory tasks automatically. Programmer doesn't need to give attention to allocate or free memory. Although Java is limited to pointer power, it isn't limited completely. Instead of pointer, programmer can use reference. It looks like pointers but it is not the same as the pointer. Programmer can manipulate memory content if he/she has enough knowledge about pointer and reference.

In C if programmer want to create an array by dynamically , he/she need to know information about pointer syntaxly. Besides this programmer need to know semantic knowledge about pointers. For instance what is the mean of the pointer , what is the content , adress . where does program give memory to pointer. What happens if memory is not enough for program. What happened if program doesnt free to allocated memory. These questions are just instance. There is too much quesitons about this topic. I mentioned the power at previous section. I have used C language and I have encountered a lot of trouble. My programms dont be safe a long time .

Java have eliminited all of these question. It have taken power from programmer and have started to use itself. java give user more suitable environment for programming and java programs are more safer than C programms

2.2 Data Types

C provides a rich set of primitive data types, including int, float, char, double, and long. These types are low-level and closely map to hardware data representations, offering a high degree of control. With these primitive data types C has Derived data types , pointers, enumeration , user definedn types and void. These data types add features to language. Thanks to these programmer can manage complex data types or create own data types. Although C is not object orinated language it can use like object orianted language. This initiative is too hard . because there are much syntax issue in this initiative.

In Java, data types are used to define the characteristics and meaning of data stored in variables. Java provides primitive data types like int, float, char, and boolean for efficient representation of integers, floating-point numbers, characters, and binary values, respectively. Additionally, there are reference data types, including classes, interfaces, enums, and arrays, which are used for defining objects, contracts for classes, sets of named constants, and collections of elements of the same type. Java also offers generics for writing generic and type-safe code, user-defined data types for custom data structures and behaviors, and wrapper classes for primitive types to provide utility methods and interact with Java collections. Finally, the void data type is used as a return type for

methods that do not return a value, indicating the absence of a return value. These data types are selected based on their intended purpose, enabling developers to write clear, efficient, and reliable code that reflects the semantics of their applications.

In conclusion C and Java has various data types . Someones are commen , someones are special. According to me Java has more easy syntax more suitable data types.if programmer dont implement the sofwaore for hardware or operating system , he / she dont need to use C language . coding C language can be punishment

2.3 Object – Orianted Features

Although C is not naturally object-oriented, it does support some object-oriented programming via the usage of function pointers and structs. Although data structures and functions that mimic objects and methods can be created, the procedure is tedious and less intuitive than in Java.

Java is an object-oriented language that supports polymorphism, encapsulation, inheritance, and classes. Because of this, it's a great option for developing big, modular systems and sharing code. Because Java encourages proper coding standards, developers that support the object-oriented paradigm will find it easier to use.

2.4 Exception Handling

There are no built-in exception handling methods in C. In C, error handling is usually accomplished via hand checks and return codes, which can result in coding that is prone to errors.A strong exception handling system is provided by Java. It enables exception handling and catching by developers, resulting in more predictable and elegant error recovery in applications.

2.5 Platform Independence

Because C code is platform-dependent, it might not function properly on one platform without additional changes. Cross-platform C code porting can be a difficult and time-consuming process.

Java is renowned for being platform-neutral. It accomplishes this by compiling to bytecode, which the JVM runs. Java is a popular choice for cross-platform applications because of its portability.

2.6 Safety and Security

Data leaks, pointer problems, buffer overflows, and other security flaws can affect C programs. The programmer must explicitly implement security methods.

Java's type safety and stringent memory management help to increase security. Common security vulnerabilities are less likely because of the runtime environment's security protections and the lack of direct memory access.

3. Other Aspects Comparison

3.1 Readability

Main purpose of readability is simplified to complexity of the code. Sometimes developer can write complex code. At this point developer needs to create clear structure. Thanks to this developers can fix issue or other developers can understand this code. Although developers need to create readable structure, programming languages shouldn't give all work to them. In this case Java and C have certain differences to increase readability.

Java is a programming language that uses principles of the object-oriented principles. Therefore Java has some rules such as encapsulation, polymorphism etc... Java is a pure object-oriented programming language. Thanks to this developers require to obey object-oriented programming principles, these principles' structure. If developers want to create a program he/she must to encapsulate data and method. This case gives developer readable code.

In contrast, C allows for greater coding style flexibility. Although this could be useful for seasoned developers, improper organization and documentation could result in less readable code. Because of its relatively flexible and succinct syntax, C can occasionally produce terse code that necessitates a deeper

comprehension of the programmer's intent. When writing C code, developers should take extra initiative to make the code legible by giving variables and functions meaningful names, leaving plenty of comments, and organizing the code logically.

In conclusion, Java's stringent grammar, object-oriented ideas, and naming rules all contribute to its readability benefits. Contrarily, C supports a wider variety of coding styles, which can be advantageous but also mean that additional work is needed to guarantee code clarity.

3.2 Writability

The ease and speed with which code can be created in a programming language is known as writability. Java is frequently commended for its object-oriented, high-level design, which makes it writable. The language simplifies coding by abstracting away a lot of minute details. A number of standard libraries in Java allow developers to save time by not having to reinvent the wheel when performing routine activities. Because it makes coding simpler, Java's emphasis on eliminating boilerplate code also improves writability. But compared to C, Java code can be verbose, meaning that developers might need to write more lines of code to accomplish certain tasks.

For veteran coders, however, C provides greater low-level control, which might be advantageous. Because of its direct control, C is an excellent choice for system-level programming, embedded systems, and other applications where efficiency is crucial. Developers can use this power to create highly efficient code for certain tasks. In contrast to Java, this degree of control may also require more work to accomplish the same functionality. Because of C's ease of use and versatility, handling memory, input/output, and data structures frequently requires creating additional code.

In the end, the writability of Java vs. C ultimately comes down to the needs of the project. Java's large libraries and high-level abstractions contribute to its writability; these are useful for quickly developing applications. Contrarily, C offers fine-grained control, allowing programmers to tailor code to particular requirements—albeit at a cost of extra work.

3.3 Reliability

In programming, reliability is the capacity to write error-free, expected-behaving code. Java has dependability as a top priority. Runtime errors are less likely because of the language's strong type checking, which helps prevent many typical programming errors. Further enhancing reliability is Java's automatic garbage collection memory management, which lowers the possibility of memory leaks and buffer overflows—common problems in languages like C.

When utilized properly, C's direct control over memory and hardware can improve reliability. Carefully controlling memory allocation and deallocation allows developers to write code that is both error-free and efficient. If not properly managed, this same degree of control could provide problems for reliability. Memory management problems in C can lead to segmentation errors and memory leaks, among other memory-related problems. C's dependability is mostly determined by the programmer's skill and thorough testing.

In conclusion, Java's automatic memory management and stringent type checking are the foundation of its dependability advantages. It is a great option for applications that require resilience and a lower chance of common programming errors. When used properly, C can be just as dependable, but it puts more responsibility on the programmer to manage memory and hardware interactions properly.

3.4 Learning Curve

A programming language's learning curve describes how simple it is for developers to become fluent in that language. The complexity and design of Java and C have an impact on their respective learning curves.

The learning curve for Java is kinder, especially for novices and people who are not familiar with programming. This is mostly because of its object-oriented, high-level design. Java makes a lot of low-level details abstract, which makes some programming tasks easier. Pointers, low-level hardware interactions, and memory management are not concerns for developers. Furthermore, a lot of typical

jobs are made simpler by Java's large standard libraries, which lowers the learning curve for achieving useful objectives.

C has a steeper learning curve than the others. It makes deeper information about memory management and hardware interactions visible. Bit-level operations, manual memory management, and pointers are all ideas that developers need to grasp. For some applications, this deeper understanding may be useful, but it also makes learning more difficult—especially for inexperienced programmers.

In the end, the developer's background and goals determine which learning curve is best. Java has a more forgiving learning curve, which is helpful for novices and people who value developing applications quickly. Those who want to learn more about embedded systems, performance-critical applications, or system-level programming may find that C has a higher learning curve.