

Programming language essay



re Programming Languages Submitted To: Madam Amna Naveed Submitted
By: The Programmers COMSATS - Lancaster Dual Degree Programme

COMSATS Institute of Information Technology Lahore PROGRAMMING

LANGUAGES SUBMITTED TO: Madam Amna Naveed Lecturer Department of
Humanities & Social Sciences COMSATS, Lahore SUBMITTED BY: Sana

Naheed Mohammad Nawal Shaukat Syed Usama Nasir Hafiz Abdul Shakoor

ddp-fa12-bse-115 ddp-fa12-bse-103 ddp-fa12-bse-130 ddp-fa12-bse-034

Date: April 18, 2013 COMSATS - Lancaster Dual Degree Programme

COMSATS Institute of Information Technology Lahore 17th April 2013

Madam Amna Naveed Lecturer CIIT Lahore, Lahore Subject: Report on

Programming Languages Dear Madam, We are submitting herewith our

report entitled " Programming Languages" as a partial fulfillment of the

Report Writing Skills course requirement. The main purpose of this report is

to promote the knowledge about the different languages used in the industry

The report states the brief history of the language and its basic syntax. Some

of the distinguishing features which differentiate it from other languages

have also been described. Also discussed are the limitations of the language

and the amendments made in them over the years.

As we have just started programming we did not know much about the

languages beforehand. So during the research phase, we learnt a lot new of

things about the languages. But we could not find any resource person

having knowledge pertaining to the languages we chose. So we had to rely

completely on the secondary sources. We hope that this report will merit

your approval. Sincerely, The Programmers ? ? ? ? Sana Naheed Mohammad

Nawal Shaukat Syed Usama Nasir Hafiz Abdul Shakoor ddp-fa12-bse-115

ddp-fa12-bse-103 ddp-fa12-bse-130 ddp-fa12-bse-034 iii ABSTRACT

Authorized by Madam Amna Naveed, Lecturer in COMSATS Lahore, this report is submitted on 17th April 2013. The main purpose of writing the report is to promote and increase the knowledge about the basic programming languages. Software engineering students need to master the skills of programming not only for getting their degree but also for their future career. A programming language is a manmade language designed to express the calculations can be performed by a machine, including a computer. Programming languages creates several programs that control the behavior of a machine to express any form of human communication assigned.

The theory of programming languages is a branch of computer science deals with the design, implementation, analysis, characterization and classification of programming languages and their individual characteristics. The introduction, brief history, basic syntax, versions and limitations have been discussed and positive conclusions have been drawn. The data was primarily collected from secondary sources e. g. from books and from various websites of the internet. The conclusion of the study is that the programming languages have immense importance in the life of a programmer.

Every language has its unique features which distinguish it as well as limitations which lead to continuous amendments. iv CONTENTS TABLE OF CONTENTS LETTER OF TRANSMITTAL

..... iii ABSTRACT

..... iv 1.

INTRODUCTION.....

..... 1 2. INTRODUCTION TO HTML

..... Introduction

..... 3 2. 1 Brief History

..... 4 2. 2

HTML Basic Document

..... 4 2. 3 Versions of

HTML 6 2. 4

Limitations and Amendments

..... 6 3.

PYTHON

.....

..... 8 Introduction

.....

..... 9 3. 1 Brief History

..... 10 3. 2

Python Basic Syntax

..... 10 3. 3 Versions of

Python 1 3. 4

Important Features

..... 12 3. 5

Limitations

..... 13 3. 6 Salient Features With Other Languages

..... 13 4. ASSEMBLY LANGUAGE

..... 14 Introduction

.....

..... 5 4. 1 Brief History

..... 16 4. 2

Basic Concepts

..... 16 i. ii. iii.

iv. Registers.....

16 Equates

18 The mov Instruction 8

Declaring a String 19 4.

3 Basic Document

..... 19 iv

CONTENTS 4. 4 Limitations

..... 20 4. 5

Current Usage

..... 20 5.

INTRODUCTION TO JAVA

..... 2 Introduction

.....

..... 23 5. 1 Brief History

..... 24 5. 2

Basic Java Document

..... 25 5. 3 Java Virtual

Machine 26 5. 4

Features of Java

..... 6 5. 5

Limitations

..... 28 5. 6

Comparison Between Java and C++

..... 29 SUMMARY

.... 31 SUGGESTIONS

..... 32

BIBLIOGRAPHY

..... 4

GLOSSARY

... 36 INDEX

..... 39 v CONTENTS LIST OF FIGURES A. Html Code For Printing Hello

World 6 B. Python

Logo

..... 9 C. Guido Van Rossum (Creator Of

Python) 10 D. Python Code

For Printing Hello World

11 E. 32-Bit General Purpose Register

..... 17 F. Memory

Models

..... 18

G. Assembly Code For Printing Hello World

..... 9 H. Java Logo

..... 23 I. Java Code For Printing Hello World

..... 25 J. JVM

..... 26 LIST OF TABLES

..... 24 2.

Comparison of C++ and Java

..... 29 vi CHAPTER

1 INTRODUCTION 1 INTRODUCTION This report was assigned by Madam Amna Naveed, Lecturer in COMSATS Lahore, to be submitted by 17th April 2013. The report has been written due to the colossal importance of the programming languages and to promote the knowledge about them. In order for computers to accept commands from humans and perform tasks vital to productivity and e-commerce, a means of communication must exist. Programming languages provide this necessary link between man and machine.

Because they are quite simple compared to human language, rarely containing more than few hundred distinct words, programming languages must contain very specific instructions. There are more than 2,000 different programming languages in existence, although most programs are written in one of several popular languages, like BASIC, COBOL, C++, or Java.

Programming languages have different strengths and weaknesses.

Depending on the kind of program being written, the computer it will run on, the experience of the programmer, and the way in which the program will be used, the suitability of one programming language over another will vary.

This report will be analyzing four important programming languages namely HTML, Python, Assembly and Java. All the languages have been established according to the following outline: ? ? ? ? ? ? Introduction Brief History Basic Layout / Syntax of the Language Unique Features Versions of the Language Limitations and Amendments 1 See Glossary 2 CHAPTER 2 INTRODUCTION TO HTML Chapter No 2 Introduction to HTML INTRODUCTION Hyper Text Markup Language (HTML) is the main markup language for creating webpages and other information that can be displayed in a web browser.

Hypertext Markup Language, a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World. ? ? ? ? Hyper is the opposite of linear. Old-fashioned computer programs were necessarily linear - that is, they had a specific order. But with a "hyper" language such as HTML, the user can go anywhere on the web page at any time. Text is just what you're looking at now - English characters used to make up ordinary words. Mark-up is what is done to the text to change its appearance. For instance, "marking up" your text with **before it and after it** will put that text in bold.

Language is just that. HTML is the language that computers read in order to understand web pages. Developed by scientist Tim Berners-Lee in 1990, HTML is the "hidden" code that helps us communicate with others on the

World Wide Web (WWW). The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms.

It provides a means to create structured documents by denoting structural semantics² for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. HTML coding is what makes your website look like a website, rather than just plain text. With HTML, you can: ? ? ? ? ? ? Create titles, headings and subheadings. Make a bulleted or numbered list. Delineate paragraphs. Put text in bold or italics. Add pictures and links to your site. And much more. ² See Glossary 4 Chapter No 2

Introduction to HTML 2. Brief History In 1980, physicist Tim Berners-Lee, who was a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners-Lee wrote a memo proposing an Internet-based hypertext system. Berners-Lee specified HTML and wrote the browser and server software in the last part of 1990. In that year, Berners-Lee and CERN data systems engineer Robert Cailliau collaborated on a joint request for funding, but the project was not formally adopted by CERN. In his personal notes from 1990 he listed “ some of the many areas in which hypertext is used” and put an encyclopedia first.

The first publicly available description of HTML was a document called “ HTML Tags”, first mentioned on the Internet by Berners-Lee in late 1991. It <https://assignbuster.com/programming-language-essay/>

describes 18 elements comprising the initial, relatively simple design of HTML. Eleven of these elements still exist in HTML 4. Tim Berners-Lee was the primary author of HTML, assisted by his colleagues at CERN, laid the foundation of an international scientific organization based in Geneva, Switzerland. Tim Berners-Lee is currently the Director of the World Wide Web Consortium, the group that sets technical standards for the Web. . 2 HTML Basic Document Using the correct HTML document structure when creating a web page is important. If the HTML document structure is incorrect the web page can break or the search engine spider 3 may not be able to read the page. The DOCTYPE declaration is the first part of coding that you should enter in your HTML document. This is required if you wish to validate your document with the W3C's validation service. Web browsers need to know what version of HTML/XHTML your page is written in to process the code correctly. 3 See Glossary 5 Chapter No 2 Introduction to HTML

The HTML tags and are present in all HTML documents. These tags identify the document's contents as HTML to the browser. The tag goes in the line right under your DOCTYPE declaration. is the last line of coding in your document. Opening html tag: The head tags, the and tags identify the document's head area. The information between these two tags is not visible on your page. Opening head tag: The title tag creates the page title that is seen in the title bar of the web page. Title of the document The closing head tag defines the end of the document's head section.

The body tags surround the body (contents) of your web page. The body of the document The closing HTML tag is the last line in your HTML document. Don't put anything after this tag! Your page will not validate if you do. The <https://assignbuster.com/programming-language-essay/>

body of the document 6 Chapter No 2 Introduction to HTML Fig A: Code for printing " Hello World" 2. 3 Versions of HTML HTML is an evolving language, and each new version is given a number. The various versions of HTML along with year of release are as: ? ? ? ? HTML 1. 0 (1989 - 1994) HTML 2. 0 (1995) HTML 3. 20 (1997) HTML 4. 01 (1999) However, the evolution of HTML has now ceased — HTML 4. 1 is the last version of HTML. For the future, HTML is being replaced by a new language, called XHTML — for the extensible Hyper Text Markup Language. XHTML Versions XHTML is a separate language that began as a reformulation of HTML 4. 01. It continues to be developed: ? ? ? ? XHTML 1. 0, published January 26, 2000, as a W3C4 Recommendation, later revised and republished August 1, 2002 XHTML 1. 1 published May 31, 2001, as a W3C Recommendation. XHTML 2. 0 was a working draft, but work on it was abandoned in 2009 in favor of work on HTML5 and XHTML5. XHTML5, which is an update to XHTML 1. , is being defined alongside HTML5 in the HTML5 draft. 4 See Glossary 7 Chapter No 2 2. 4 Limitations and Amendments in HTML Introduction to HTML With the advancement of the internet, HTML versions began to certain types of limitations. This lead to the advancement of the language with newer version, each of which was designed to make up for the lacks in the previous version. HTML 1. 0 (1989 - 1994) The first version of HTML that supported inline images and text controls. HTML 1. 0 was very limited in terms of styling and presentation of content. In HTML 1. 0, for example, you could not: ? ? ? use tables or frames, specify fonts, change page background, or use forms Because of these limitations, every web page created with HTML 1. 0 looked the same with similar background and the type of font used. HTML 2. 0 (1995) This specification supported more browsers. HTML 2. 0 was considerably

improved to support: It also supported: ? ? ? forms with limited set of form elements such as text boxes, and option buttons change of page background use of tables HTML 3. 20 (1997) This version included support for creating tables and expanded options for form elements.

This version also allowed web pages to include complex mathematical equations. HTML 4. 01 (1999) This version added support for style sheets and scripting ability for multimedia elements. HTML 4. 01 focused on separating presentation styling information from the actual content by the use of style sheets as HTML 3. 20 resulted in difficult maintenance because presentation styling information was included directly in the webpage. In HTML 4. 0 with the use of style sheets, it is now possible to change the appearance/look of the website by changing just the style sheet (s) itself.

In comparison, in the earlier versions of HTML making the same changes for the entire website meant changing the styling information in the individual pages. 8 CHAPTER 3 PYTHON Chapter No 3 Python INTRODUCTION Python is an interpreted language widely used, one of the easiest to learn and use. A day is enough to start programming with Python and you can write scripts in a few hours while you need for days with other languages . It has powerful features such as lists, tuples (ordered list of elements) dictionaries that allow you to translate your ideas into lines of code directly.

The lists are integrated and made it a successor to the Awk and any other word processor language. There is a version for Windows, Linux and other platforms. Fig B: Python Logo Python is a general-purpose, high level programming language⁵ whose design philosophy emphasizes code

readability. Python's syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as "C", and the language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple "programming paradigms" 6, a fundamental style of computer programming.

It features a fully dynamic type system and automatic memory management and has a large and comprehensive standard library. Like other "dynamic languages", Python is often used as a "scripting language", but is also used in a wide range of non-scripting contexts. Using third-party tools, Python code can be packaged into standalone executable programs. Python interpreters are available for many operating systems. 5 6 See Glossary See Glossary 10 Chapter No 3 Python 3. 1 Brief History The Developer of the language, "Guido Van Rossum", was Google employee from 2005 to 2012 and works for Dropbox since.

Python was created in the late 1980s and its implementation was started in December 1989 by "Guido van Rossum" at CWI (Centrum Wiskunde & Informatica, National Research Institute for Mathematics and Computer Science) in the Netherlands as a successor to the ABC language. Fig C: Guido van Rossum (Creator of Python) "Van Rossum" is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, Benevolent Dictator for Life (BDFL). 3. 2 Python's Basic Syntax Python has a broad range of basic data types⁷, basic building blocks.

The python language has many similarities with “ C” and “ JAVA”. However, there are some definite differences between the languages. Python was designed to be a highly readable language. Consider a simple Python program in a script. 7 See Glossary 11 Chapter No 3 Python Printing Code: All python files will have extension “. py”. So put the following source code in a test. py file. print “ Hello, Python! ”; Now try to run this program \$ python test. py This will produce following result: Hello, Python! Fig D: Code for printing “ Hello Python” 3. Versions of Python Python is an evolving language having limitations. The limitations lead to the release of a better version. Each new version is given a number. The versions along with the main amendments made are: ? ? ? ? ? 2. 2 2. 3 2. 4 2. 6 3. 0 (First version) (Addition of new keywords) (Added support for the optional key argument) (Return a new featureless object. object is a base for all new style classes) (Changing in syntax) New Version ? Python 2. 0 was released on 16 October 2000, with many major new features including a full garbage collector and support for Unicode8.

With this release the development process was changed and became more transparent and communitybacked. ? Python 3. 0 (also called Python 3000 or py3k) was released on 3 December 2008 after a long period of testing. Many of its major features have been back ported to the backwards-compatible Python 2. 6 and 2. 7. 8 See Glossary 12 Chapter No 3 Python 3. 4 Important Features Python has some unique features which differentiates it from the other languages. Some of which are listed below: ? Python is an object oriented 9(that represents concepts as objects), interpreted language 10 a language that is executed by coder directly). ? ? ? ? ? Variables are

dynamic, the type is not declared and may change. Indentation is used for block recognition and this is unique to Python. Tuples are variables or object packed all together, for functions return. Lists and dictionaries are other built-in composed objects. Functions may be embedded inside other functions. Can be extended with C modules. 3. 5 Limitations Python has some serious drawbacks which need to be focused on in the upcoming versions. Some of the faults complained by the coders are: ? ? ? ? ? ?

Code execution is slow. This is why it tends to be replaced by Go. You cannot declare a variable like in JavaScript with var. This reduces clarity and security of the code. The object orientation is rather rustic and improvised. Using indentation to define a block is a design choice often contested. So if you comment a line that changes the structure. Some Built-in global functions should be object methods instead (but PHP (a programming language) is much worse in this regard). Python 3. 0 changes the syntax of the language which makes it partially incompatible with the previous ones.

Due to which the previous codes become null and void e. g. : o print x is replaced by print (x). o Two types of string, data and str not compatible. o Lists are replaced by views and iterators. o 9 10 See Glossary See Glossary 13 Chapter No 3 Python 3. 6 Salient Features with Other Languages Python's statements include many statements which are also used in other major languages e. g. Java, C, C++. This increases the portability of the program (to make a program general enough to work on different types of systems). Some statements are listed below: ? ? ? ? ?

The if statement, which conditionally executes a block with else and elif (a contraction of else-if). of code, along The for statement, which iterates over an alterable object, capturing each element to a local variable for use by the attached block. The while statement, which executes a block of code as long as its condition is true. The class statement, which executes a block of code and attaches its local namespace to a class, for use in “ object-oriented programming”. The def statement, which defines a “ function” or “ method”.

14 CHAPTER 4 ASSEMBLY LANGUAGE Chapter No 4

Assembly Language INTRODUCTION The assembly language is a low-level¹¹ (that provides little or no abstraction from a computer’s instruction set architecture) way to instruct a computer to carry out a task. Computers do not inherently understand instructions from people. At the most basic level, computers understand instructions in binary language, that is, sequences of zeroes and ones. Binary language or machine language is extremely cumbersome to program in. Assembly language was invented as a symbolic representation of the underlying sequences of zeroes and ones.

Suppose we were to program a game which, upon a certain action, awards the player with five points. The computer keeps the score in a certain location; the locations are either a storage place within the central processing unit (CPU), called a “ register,” or in some space in the memory. The computer understands locations and low-level actions called operations. At the most basic level, computer hardware is controlled through machine language, consisting of numbers (mainly zeros and ones). Immediately above machine languages are assembly languages, which use mnemonic names instead of numbers to represent instructions.

This level of language is the lowest a programmer is likely to see. Special programs known as assemblers take assembly code and translate it into the machine language used by a computer's hardware. Although they aren't numeric, assembly languages have several disadvantages, including the fact that they are hard to understand and often are very specific to a certain machine's CPU. Each kind of CPU has its own form of assembly language. In the early 2000s, programs usually were not written directly in assembly language. Rather, it was used by experienced programmers to work on critical parts of computer programs. 1 See Glossary 16 Chapter No 4 Assembly Language 4. 1 Brief History The first assembler¹², assembly language translator, appears in the early mainframe computer EDSAC in late 1940s. Electronic Delay Storage Automatic Calculator (EDSAC) was an early British computer. The machine, having been inspired by John von Neumann's seminal First Draft of a Report on the EDVAC, was constructed by Maurice Wilkes and his team at the University of Cambridge Mathematical Laboratory in England. EDSAC was the first practical stored-program electronic computer.

EDSAC ran its first programs on 6 May 1949, when it calculated a table of squares and a list of prime numbers. The initial orders were hard-wired on a set of unis selector switches and loaded into the low words of memory at start up. By May 1949, the initial orders provided a primitive relocating assembler taking advantage of the mnemonic design, all in 31 words. That was the world's first assembler, and arguably the start of the global software industry. After this, many assemblers appeared on various mainframes,

including Regional Assembly Language assembler, Whirlwind assembler and Rochester assembler and many others.

Operating systems were almost exclusively written in assembly language until the widespread acceptance of C in the 1970s and early 1980s. Many commercial applications were written in assembly language as well, including a large amount of the IBM mainframe software written by large corporations. COBOL and FORTRAN eventually displaced much of this work, although a number of large organizations retained assembly-language application infrastructures well into the 90s. In the 1980s, TASM and MASM were developed by Borland and Microsoft.

Later, new features were added to assembly language. Some assemblers have incorporated structured programming elements to encode execution flow. A new generation of assembly language appears these years named "typed assembly language" on which assembly language is extended.

4. 2 Basic Concepts

a) Registers

Modern (i. e. 386 and beyond) x86 processors have eight 32-bit general purpose registers, as depicted in Figure 4. 1. Registers are available to be used as stores for binary data. Collectively the data and address registers are called the general registers.

2 See Glossary

17 Chapter No 4 Assembly Language

The register names are mostly historical. For example, EAX used to be called the accumulator since it was used by a number of arithmetic operations, and ECX was known as the counter since it was used to hold a loop index. Whereas most of the registers have lost their special purposes in the modern instruction set, by convention, two are reserved for special purposes — the stack pointer (ESP) and the base

pointer (EBP). For the EAX, EBX, ECX, and EDX registers, subsections may be used.

For example, the least significant 2 bytes¹³ (the rightmost ones) of EAX can be treated as a 16-bit register called AX. The least significant byte of AX can be used as a single 8-bit register called AL, while the most significant byte of AX can be used as a single 8-bit register called AH. These names refer to the same physical register. When a two-byte quantity is placed into DX, the update affects the value of DH, DL, and EDX. These sub-registers are mainly hold-overs from older, 16-bit versions of the instruction set. However, they are sometimes convenient when dealing with data that are smaller than 32-bits (e. . 1-byte ASCII characters). When referring to registers in assembly language, the names are not case-sensitive. For example, the names EAX and eax refer to the same register. Fig E: 32-bit General Purpose Register 13

See Glossary 18 Chapter No 4 Assembly Language Types of Registers Index registers: The main registers are 16-bit registers. There are four general purpose registers: AX, BX, CX and DX. They are split up into 8-bit registers. AX is split up into AH which contains the high byte and AL which contains the low byte.

On 386's and above there are also 32-bit registers, these have the same names as the 16bit registers but with an ' E' in front i. e. EAX. You can use AL, AH, AX and EAX separately and treat them as separate registers for some tasks. On 386's and above there are also 32-bit index registers: EDI and ESI.

b) Memory Models A memory model describes the interactions of threads through memory and their shared use of the data. There are several memory

models available. In standalone assembly language programming, the small is usually the best choice. In the table below those models are described.

Fig F: Memory Models c) Equates In assembly language, constant values are known as equates, referring to the EQU directive, and associates values with identifiers such as " Base Address" and " Max Value". It is important to note that equates may appear anywhere in the program without restriction. Some examples are given below: ? ? ? after declaring a symbol with EQU you cannot change its associated value, the values declared with "=" can be changed as often as you want, EQU can declare all kinds of equates including numbers, expressions and characters 19 Chapter No 4) The Mov Instruction Assembly Language The instruction used to move an operand from one location to another one is MOV. It can be used to address general purpose register or to address memory locations. Its format is that the latter is moved into the former after the mov keyword. Example: mov ax 15 ; puts 15 into ax e) Declaring a String To declare a string, a sequence of characters, we are going to type: Message DB " Hello World! \$" The string is terminated with a dollar sign. DB is short for declare byte and the message is an array of bytes (an ASCII character takes up one byte).

Data can be declared in a number of sizes, bytes (DB) and words (DW). 4. 3 Basic Document For printing plain text on the output, input the following code: . model small . stack 100h . data msg db ' Hello world! \$' . code start: mov ah, 09h ; Display the message lea dx, msg int 21h mov ax, 4C00h ; Terminate the executable int 21h end start Fig G: Code for Printing " Hello world! " The output will be as: 20 Chapter No 4 Hello world! Assembly Language Being a low level language, it requires more and complex coding <https://assignbuster.com/programming-language-essay/>

for tasks. The same printing text task may be done much easily in the other languages e. . Java, C, and C++ etc.

4. 4 Limitations of Assembly Language

Assembly language being a first generation or low level language has many limitations. These drawbacks ultimately led to the evolution of programming languages. Some of the downsides of the language are as following: ? ? ? ? ?

No Symbolic names for memory locations. You need to keep track of the exact memory location that a piece of data is stored. That is, you must manipulate memory locations directly. Hard to read. Although a few improvements have been made by eliminating hex code, the command names are not always clear.

Code is still machine dependent. It has not really moved that far away from the machine language - just pseudo-English labels have been placed on it. Hard to maintain and debug¹⁴. Finding mistakes in machine code is difficult. Correcting them or adding new features can also be a challenge. Code must be heavily documented. It's very difficult (if not impossible) to figure out what a program does by reading the code. Detailed explanation must be prepared for future coders (including the original programmer) who need to modify or use the code for every machine.

4. Current Usage

There have always been debates over the usefulness and performance of assembly language relative to high-level languages. Assembly language has specific niche uses where it is important. There are some situations in which developers might choose to use assembly language: ? ? A stand-alone executable of compact size is required that must execute without recourse to the run-time components or libraries associated with a high-level language Code that must interact directly with the hardware, for example in

device drivers and interrupt handlers 14 See Glossary 21 Chapter No

4 ? ? ? ? ? ?

Assembly Language Programs that need to use processor-specific instructions not implemented in a compiler¹⁵ Situations where no high-level language exists, on a new or specialized processor Programs that need precise timing Situations where complete control over the environment is required, in extremely high security situations where nothing can be taken for granted. Computer viruses, boot loaders, certain device drivers, or other items very close to the hardware or low-level operating system. Instruction set simulators for monitoring, tracing and debugging where additional overhead is kept to a minimum. 5 See Glossary 22 CHAPTER 5

INTRODUCTION TO JAVA Chapter No 5 Introduction to Java INTRODUCTION

Java is a high-level language (a type of programming language which is designed to be easy for humans to read) like C, FORTRAN, Smalltalk, Perl, and many others. It made the promise of “ Write Once, Run Anywhere”, with free runtimes on popular platforms. It was fairly secure and its security was configurable, allowing for network and file access to be limited. You can use Java to write computer applications that play games, store data or do any of the thousands of other things computer software can do.

Compared to other programming languages, Java is most similar to C.

However although Java shares much of C's syntax, it is not C. Knowing how to program in C or, better yet, C++, will certainly help you to learn Java more quickly, but you don't need to know C to learn Java. Fig H: Java Logo . Unlike conventional languages which are generally designed either to be compiled to native (machine) code, or to be interpreted from source code at runtime,

<https://assignbuster.com/programming-language-essay/>

Java is intended to be compiled to a byte code, which is then run by a Java Virtual Machine.

What's most special about Java in relation to other programming languages is that it lets you write special programs called applets¹⁶ that can be downloaded from the Internet and played safely within a web browser. Java language is called as an Object-Oriented Programming language. ¹⁶ See Glossary 24 Chapter No 5 Introduction to Java 5. 1 Brief History Java is an object-oriented programming language developed by James Gosling and colleagues at Sun Microsystems in the early 1990s. Java was started as a project called " Oak" by James Gosling in June 1991.

Gosling's goals were to implement a virtual machine and a language that had a familiar C-like notation but with greater uniformity and simplicity than C/C++. The first public implementation was Java 1. 0 in 1995. New versions for large and small platforms (J2EE and J2ME) soon were designed with the advent of " Java 2". Sun (Stanford university network) has not announced any plans for a " Java 3". Year 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2002 2004 Progress Sun decided to developed software that could be used for electronic devices.

And the project called as Green Project head by James Gosling.

Announcement of a new language named ? Oak? The team verified the application of their new language to manage a list of home appliances using a hand held device. The World Wide Web appeared on the Internet and transformed the text-based interface to a graphical rich environment. The team developed a new Web browser called ? Hot Java? to locate and run

Applets. Oak was renamed to Java, as it did not survive ? legal? registration. Many companies such as Netscape and Microsoft announced their support for Java.

Java language is now famous for Internet programming as well as a general purpose OO language. Sun releases Java Development Kit(JDK 1. 1) Sun releases Software Development Kit (SDK 1. 2) Sun releases Java 2 platform Standard Edition (J2SE) and Enterprise Edition (J2EE). J2SE with SDK 1. 3 was released. J2SE with SDK 1. 4 was released. J2SE with JDK 5. 0 was released.

Table 1: History of Java 25 Chapter No 5 Primary Goals Introduction to Java

There were five primary goals in the creation of the Java language: ? ? ? ? ? It should use the object-oriented programming methodology.

It should allow the same program to be executed on multiple operating systems. It should contain built-in support for using computer networks. It should be designed to execute code from remote sources securely. It should be easy to use by selecting what was considered the good parts of other object-oriented languages. 5. 2 Basic Java Document To write a java program: ? ? First of all we have to make a class in which we will write our code. Then there is a main statement in every class which is required to execute or run or to get an output of a program. This is a simple program just to print a statement Hello World.

In main we will write a statement and each statement will end with a semicolon. Both class and main will have starting and ending braces. public class HelloWorld{ public static void main(String args[]){ System. out. println(" Hello World! "); } } Fig 1: Code for printing " Hello World" This when

compiled by the Java Virtual Machine (JVM) will produce the following output

```
Hello World! public static void main(String args[]){ System. out. println("
```

```
Hello World! "); 26 } } Chapter No 5 5. 3 Java Virtual Machine (JVM)
```

Introduction to Java All the programming language compilers convert the source code to machine code.

The same job done by Java Compiler to run a Java program, but the difference is that Java compiler convert the source code into Intermediate code called byte code. This machine is called the Java Virtual machine and it exists only inside the computer memory. Following figure shows the process of compilation of a Java code by the Java Virtual Machine. Java Program Java Compiler Virtual Machine Source Code Byte Code Fig J (a): Compilation of a Java Program into Byte Code The Virtual machine code is not machine specific. The machine specific code is generated.

By Java interpreter by acting as an intermediary between the virtual machine and real machines shown below Byte Code Java Interpreter Machine code Virtual machine Real Machine Fig J (b): Compilation of a Java Program into Machine Code 5. 4 Features of Java Java has many unique features some of which are as: Java Is Simple: There are various features that makes the java as a simple language because Java is easy to learn and developed by taking the best features from other languages mainly like C and C++. It is very easy to learn Java who have knowledge of object oriented programming concepts 27 Chapter No 5

Introduction to Java Java Is Platform Independent: Java provides the facility to " Write once Run anywhere"(Known as platform independent). No language

is completely idle to this feature but java is the closest. Java provides the facility of cross-platform programs (which may be used in various systems and environments) by compiling in intermediate code known as byte code. This byte code can be interpreted on any system which have Java Virtual Machine (JVM). Java is Object-oriented: The object oriented language must support the characteristics of the OOPs and Java is a fully object oriented language.

In the Java everything is treated as objects to which functions are applied. Java is a fully object oriented language because object is at the outer most level of data structure in java Java is distributed: The widely used protocols like HTTP (Hypertext Transfer Protocol) and FTP (File Transfer Protocol) are developed in java. Internet programmers can call functions on these protocols and can get access the files from any remote machine on the internet rather than writing codes on their local system. Java is Secure: Java does not use memory pointers¹⁷ explicitly.

All the programs in java are run under an area known as the sand box. The byte code Verifier checks the classes after loading by the following criteria: ? ? No memory pointers Programs runs inside the virtual machine sandbox. Array index limit checking Java is compiled and interpreted: We all know that in Java code is compiled to byte codes that are interpreted by Java virtual machines (JVM). This provides portability to any machine for which a virtual machine has been written. The two steps of compilation and interpretation allow for extensive code checking and improved security.

Java is Robust: Java has the strong memory allocation and automatic garbage collection mechanism. It carries out type checking at both compile and runtime making sure that every data structure has been clearly defined and typed. ? ? ? Compiler checks the program for any error Interpreter checks any run time error that every data structure is clearly defined and typed Java manages the memory automatically by using an automatic garbage collector Java is Portable: The feature “ write once -run anywhere” makes java portable. Many type of computers and operating systems are used for programs.

By porting an interpreter for the Java Virtual Machine to any computer hardware/operating system, one is assured that all code compiled for it will run on that system. This forms the basis for Java’s portability. 17 See Glossary 28 Chapter No 5 Introduction to Java 5. 5 Limitations of Java Here are few limitations and disadvantages of java language because the language is still evolving. Slow: Internet as a whole ? ? ? ? Potential huge show-stopper problem. Bandwidth¹⁸ is a limited resource that is easily squandered. Minimal incentive for people to conserve. Potential demand is unbounded, potential supply is bounded.

Some solutions which may be helpful are: ? ? ? Write applications instead of applets (but then you lose some advantages) Cache your applets (only partial solution). Be a good Internet citizen: Minimize your use of detailed images and other bandwidthhungry objects Java Compiler: ? ? Not well optimized yet Takes much time to compile a large program than a good C++ compiler Java Programs: ? ? ? Byte code interpretation is 20 - 50 times

slower than C++ Much run-time error checking done Memory management is somewhat expensive Environments Limitations: Currently lacks: ? ? ?

Browser (This might be the biggest weakness) Interactive source-level debugger User interface¹⁹ builder¹⁸ See Glossary²⁹ Chapter No 5 Platform Limitations: ? Full environment available only for: o Windows 95/NT o Macintosh o Sun Solaris ? Other platforms must wait for 3rd party support Introduction to Java General stability concerns: There are many bugs in: ? ? Browsers Example programs 5. 6 Comparison between C++ and Java Java is similar to C. However although Java shares much of C's syntax. Knowing how to program in C++, will certainly help you to learn Java more quickly, but you don't need to know C to learn Java.

Here are few difference between java and C++: Java Java is true Object-oriented language. Java does not support operator overloading. It supports labels with loops and statement blocks Java does not have template classes as in C++. Java compiled into byte code for the Java Virtual Machine. The source code is independent on operating system. Java does not support multiple inheritance of classes but it supports interface. C++ C++ is basically C with Object-oriented extension. C++ supports operator overloading. It supports goto statement. C++ has template classes.

Source code can be written to be platform independent and written to take advantage of platform. C++ typically compiled into machine code. C++ supports multiple inheritance of classes. 1 2 3 4 5 6 19 See Glossary 30 Chapter No 5 7 8 Introduction to Java Runs in a protected virtual Exposes low-level system facilities. machine. Java does not support global C++

support global variable. variable. Every variable should declare in class. Java does not use pointer. C++ uses pointer. It Strictly enforces an object It Allows both procedural programming oriented programming paradigm. nd object-oriented programming. There are no header files in Java. We have to use header file in C++. Table 2: Comparison of C++ and Java 9 10 11 31

SUMMARY In order for computers to accept commands from humans and perform tasks vital to productivity and e-commerce, a means of communication must exist. Programming languages provide this necessary link between man and machine. Because they are quite simple compared to human language, rarely containing more than few hundred distinct words, programming languages must contain very specific instructions.

HTML is the main markup language for creating webpages. HTML is the “concealed” code that helps us interconnect with others people on the World Wide Web (WWW). Tim Berners-Lee was the prime author of HTML. HTML coding is what makes gives the website a nice look, rather than being just plain text. The basic HTML document consists of the doctype declaration, tags and body. The language still has some problems. To improve the language six different versions of HTML have been launched so far. XHTML is a new language that began as a reformulation of HTML 4. 01. It carries on to be established.

Python is an interpreted, object-oriented, high-level programming language. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive. It is very useful in scripting. It is simple and easy to learn its syntax . It is comprehensive language than other languages. It is more compatible in computers than others languages

<https://assignbuster.com/programming-language-essay/>

Assembly language is the mother of all the programming languages. It controls the whole hardware of computer. All languages are dependent of Assembly Language because this language provides the environment to other languages to be compiled.

All languages first go to the Assembly Language, then they are compiled and allowed to run. Java is the one of the important languages of programming. It is general-purpose, concurrent, class-based, object oriented. It is used to make games, applications, software etc. Its programs require a machine to be executed which is java virtual machine. It is intended to let application developers “ write once, run anywhere” (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another.

The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them. 32 SUGGESTIONS HTML: The following improvements may be made for the improvement of the language: ? ? ? ?

PYTHON: Syntactically Python is a power language. But quite some improvements are possible in its implementation. ? ? ? Better concurrency support. GIL (the global interpreter lock), is a mutex that prevents multiple native threads from executing Python byte codes at once. It has to be removed. Improve garbage collection.

Cleanup APIs (application programming interface) Mathematical calculations should be supported HTML should have the ability to create custom window sizes, compress files, and other standard navigational controls. It should be self-supporting rather than depending on other languages for its completion The compatibility problems should be resolved ASSEMBLY: Assembly

language is the most important language in all the programming languages, but the most important improvements which may be made is to make the syntax simpler and easy to read. As sometimes it is really difficult to understand the syntax and terms of this language.

For shorter programs like adding or subtracting , the code is easy to understand but when we talk about complex programs, it is really difficult to understand the syntax and code of the program, so if the syntax writing is made easier, it would be more helpful for the programmers to understand the code. 33 JAVA: The following developments should be done in the language: ? ? ? Java should be faster than other languages because this language is more in use than other languages. There should be another platform to run the java programs rather than virtual machine because it is too slow.

Java programs shouldn't be more memory consuming. 34 Bibliography

BIBLIOGRAPHY Assembly Language History Answers, 23 January 2008

http://wiki.answers.com/Q/History_of_assembly_language Assembly language Wikipedia, 12 April 2013 http://en.wikipedia.org/wiki/Assembly_language

“ Built-in Functions” Docs Python, April 14, 1990

<http://docs.python.org/2/library/functions.html> “ Current Limitations and

Disadvantages of Java” UOttawa, June 4, 1996 <http://www.site.uottawa.ca/~tcl/javapres/limitations.html>

FLAGS register Wikipedia 26 February 2013

http://en.wikipedia.org/wiki/FLAGS_register “ History of Java programming

language” Free Java Guide, June 4, 2009 <http://www.freejavaguide.com/history.html>

“ HTML” Wikipedia, 13 April 2013 <http://en.wikipedia.org/wiki/HTML>

“ Python Basic Syntax” Tutorials Point, January 24, 2013

<https://assignbuster.com/programming-language-essay/>

http://www.tutorialspoint.com/python/python_basic_syntax.htm " Python, for easy programming" Scriptol, April 14, 2012 <http://www.scriptol.com/programming/python.php> " Python syntax and semantics" Wikipedia, 8 April 2013 http://en.wikipedia.org/wiki/Python_syntax_and_semantics " Python (programming language)" Wikipedia, 11 April 2013 [http://en.wikipedia.org/wiki/Python_\(Programming_language\)](http://en.wikipedia.org/wiki/Python_(Programming_language)) 35 Bibliography " The HTML Document Structure" HTML Basic Tutor, July 6, 2012 <http://www.htmlbasictutor.ca/html-document-structure.htm> " The history of HTML. " About.com, April 09, 2012. <http://inventors.about.com/od/computersoftware/a/html.htm> " Versions of HTML" Utoronto, 31 January 2000 <http://www.utoronto.ca/web/HTMLdocs/NewHTML/html3.html> " Versions of HTML" Scripting master, 13 April 1999 <http://www.scriptingmaster.com/html/versions-of-HTML.asp> " What Is Assembly Language? " Wise Geek, 29 June 2003