Name:- Tanmay R. Machkar
STD:- COMP I Year: SE
Roll:- F22111017

Assignment-1

Q1. Illustrate the impact of machine architecture on programming language.

1) Hardware:-

- A computer is an integrated set of algorithms and data structures capable of storing and executing programs.
- A computer may be constructed as an actual physical device using wires, integrated circuits, circuit boards, and the like, in which case it is termed an actual computer or hardware computer.

1. Data:-
- A computer must provide various kinds of elementary data items and data structures to be manipulated.

2. Primitive operations:-
- A computer must provide a set of primitive operations useful for manipulating the data.

3. Sequence control:-
- A computer must provide mechanisms for controlling the sequence in which the primitive operations are to be executed.

4. Data access:-
- A computer must provide mechanisms to control the allocation of storage for programs and data.

5. Storage management:-
- A computer must provide mechanisms ~~for~~ to ~~controlling~~ control the allocation of storage for programs and data.

2) Firmware:-
- Common alternative to the strict hardware

realization of a computer is the firmware computer simulated by a microprogram running on a special microprogrammable hardware computer.

- Microprogram simulation of a computer is sometimes termed emulation.
- We also refer to the resulting computer as a virtual computer because it is simulated by the microprogram, without this microprogrammed simulation, the machine would not exist.

3) Software

- The simulated computer accepts as input data a program in the high-level language.
- The main simulator program performs an interpretation algorithm similar to that of decoding and executing each statement of the input program in the appropriate sequence and producing the specified output from the program.
- Host computer creates a virtual machine simulating.
- When the host computer is executing the high-level program, it is not possible to tell whether the program is being executed directly by the hardware or is first converted to the low-level machine language of the hardware computer.

Q2. List the different classes of binding times. Explain ~~th~~ with suitable examples.

Ans. The different classes of binding times are:-

1) ~~Names~~ Execution time (run time)

a. On entry to a subprogram or block.

b. At arbitary points during execution.

2) At arbitary points during execution

a. Bindings chosen by the programmer

b. Bindings chosen by the translator

3) Language implementation time

4) Language defination time

→ consider ~~on~~ the simple assignment statement

$X = X + 10$

→ Points to think:

1. Set of types of variable X
2. Type of variable x
3. Set of possible values for variable X.
4. Value of variable X
5. Representation of the constant 10.
6. Properties of the operator +.

Q3. Explain any two languages paradigms with examples.

Ans. A programming paradigm is an approach to solving programming problems.

1. Imperative languages:-

- Command driven or statement oriented
- The basic concept is machine state.
- A program consists of sequences of statement
- Execution of each instruction cause the computer to change the value of one or more location, to enter a new state.
- Syntax of such languges:-

$statement_1$;

$statement_2$;

---:---

- Many widely used Languages C, C++, Fortran, etc.

2. Applicative languages:-

- Programming language is to look at the function that the program represents rather than just the state changes as the program executes, statement by statement.
- Focus on the desired result rather than at the available data.
- The languages which emphasize this view are called applicative or functional languages.
- Syntax:-

$function_n(\ldots function_2(function_1(data))\ldots)$

Q4. List attributes of good programming languages. And explain any two in detail.

Ans. • Attributes of good programming language:-

1. ~~Ease~~ Orthogonality

- It is one of the most important feature of PL orthogonality is the property that means "Changing A does not change B".
- When the features of a language are Orthogonal, language is easier to learn and programs are easier to write because only few exceptions and special cases to be remembered.

2. Programming environment

- An appropriate programming environment adds an extra ~~facility~~ utility and make language to be implemented easily.
- The availability of - Reliable, efficient, well documentation, speeding up creation and testing by special editors, testing packages.

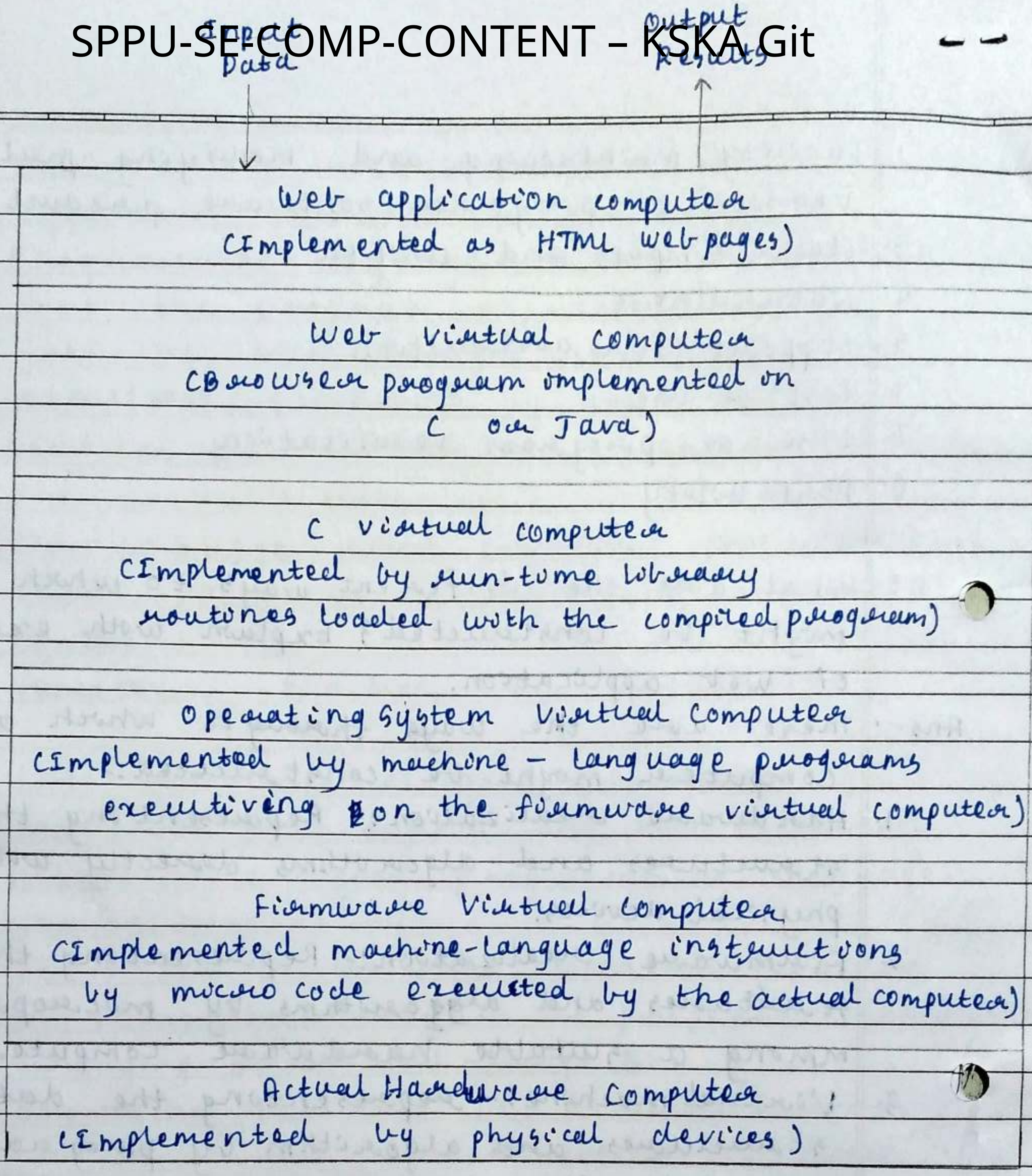
- Facility, maintaining and modifying -multi version of program software product.

3. Clear, simple and unified
4. Naturalness
5. Support for abstraction
6. Cost of use
7. Ease of program verification
8. Portability

Q5. What are the different ways to which computer might be constructed? Explain with example of web application.

Ans. • There are the ways through which a computer might be constructed:-

1. Hardware realization:- Representing the data structures and algorithms directly with physical devices.
2. Firmware realization:- Representing the data structures and algorithms by microprogramming a suitable hardware computer.
3. Virtual machine:- representing the data structures and algorithm by programs and data structures on some other programming languages.
4. Through some combination of these techniques, representing various parts of the computer directly in hardware, in microprograms, or by software simulation as appropriate.



Web application computer (Implemented as HTML web pages)

Web virtual computer (Browser program implemented on C or Java)

C virtual computer (Implemented by run-time library routines loaded with the compiled program)

Operating system virtual computer (Implemented by machine-language programs executiving on the firmware virtual computer)

Firmware virtual computer (Implemented machine-language instructions by micro code executed by the actual computer)

Actual Hardware Computer (Implemented by physical devices)

layers of virtual computers for a Web Application