**CSE 3302: Programming Languages**

**Homework 2**

**Name: Richa Pagare**

**1000 number: 1001873138**

**Date: 03/02/2022**

**1.** Describe the Difference engine.

**Ans 1:** Designed in the 1820s by Charles Babbage, the difference engine was an automatic [mechanical calculator](https://en.wikipedia.org/wiki/Mechanical_calculator) which was used to tabulate [polynomial functions](https://en.wikipedia.org/wiki/Polynomial). It was made to help solve many real-world problems like tabulating a range table for the military. This method for the difference engine was designed such that it solved complex calculations by making use of only subtraction and addition. The difference engine was way ahead of other calculators of its time. However, Babbage couldn’t complete making the difference engine due to lack of funding but it was later built in 1991 by historians.

**2.** From the following list, which can be considered as a “general purpose computer”?

a. Step Reckoner

b. Difference Engine

c. Analytical Engine

**Ans 2:** Only option c

Analytical Engine can be considered as a general purpose computer for the following reasons: -

An analytical engine has the memory to store data. Additionally, an analytical engine can also schedule multiple operations in a queue. It could be used for one computation and sequentially run the operations.

**3.** What were the limitations of Harvard Mark 1? How does current technology deal with those problem?

**Ans 3:** Harvard Mark 1 has the following limitations:

* One of the Harvard Mark 1’s biggest limitations were that it was significantly slower. This made completing calculations a time-consuming task. It used relays for circuit switching, which weren’t fast enough to compute complex problems. For example, a multiplication took 6 seconds, while division took 15 seconds.
* It consisted of approximately 3500 relays which required high maintenance since relays had to be taken care of and replaced regularly as most of the relays slowed down with time. Changing and checking faulty relays made it unreliable for solving problems.
* Other disadvantages were its loud noise while working, and its immense size, due to which various bugs and insects could easily get inside them.

Modern day computing is electronic computing to achieve faster rates of operation, decreasing the time complexity by faster circuit switching. Computers circuits now make use of on-off switches(transistors).

**4.** Why is Silicon Valley so called?

**Ans 4:** The Silicon Valley is called so as there exists a large population of industries for manufacturing transistors and semiconductors, which were created using Silicon. Silicon was thus an indispensable part of the semiconductor industry. Hence it got the name of Silicon Valley.

**5.** Make boolean logic tables for AND, OR, and XOR (A XOR B).

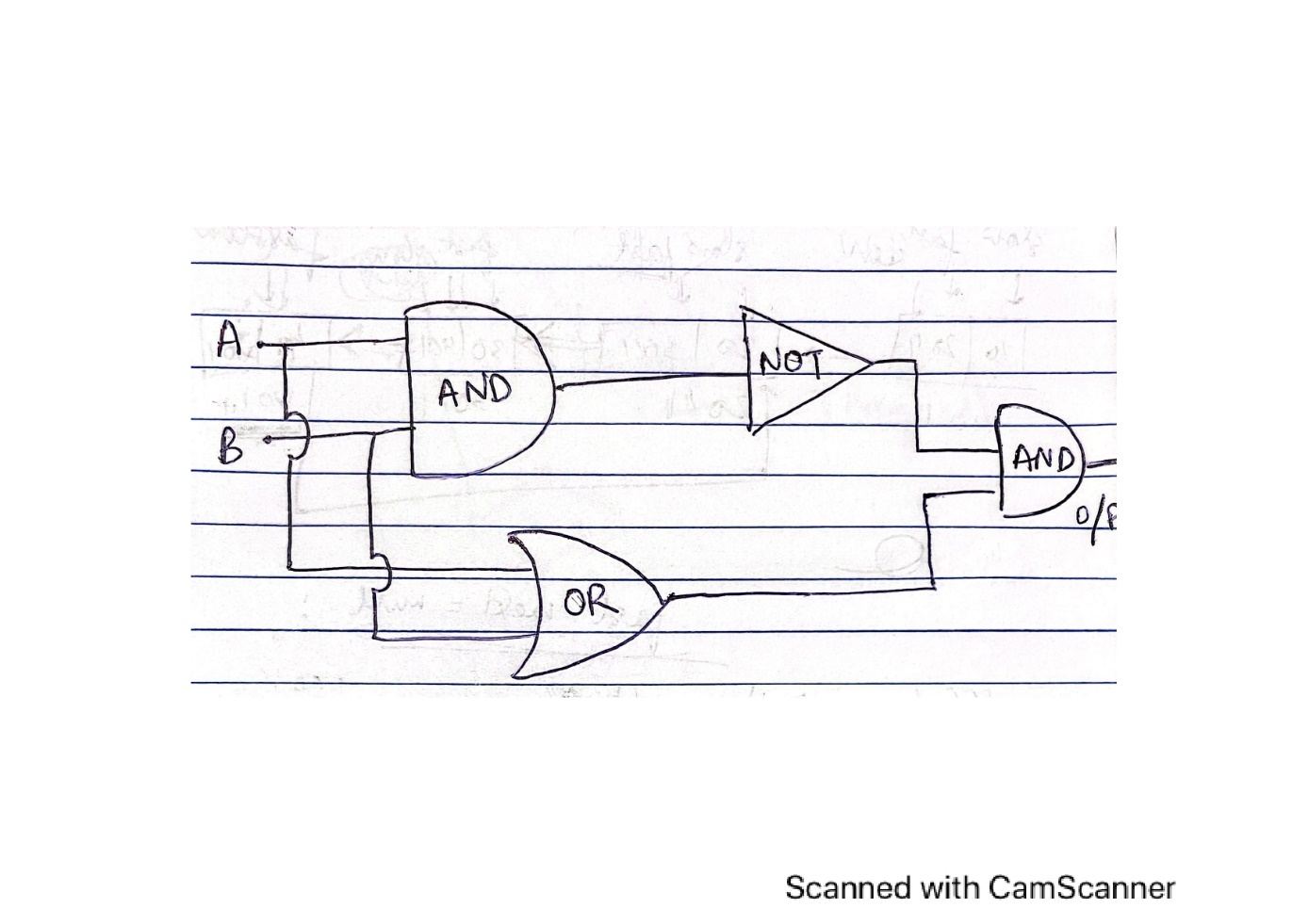
**Ans 5:** Logic Tables for AND, OR, and XOR:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **AND** | **OR** | **XOR** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |

**(1 = TRUE, 0 = FALSE)**

**6.** Can we implement XOR gate using only NOT, AND, and OR gates? If yes, how?

**Ans 6**: Yes, we can implement XOR gate using only NOT, AND, and OR gates.



**7.** How many bits would you need to represent 2018 in binary? How many bytes is that?

**Ans 7:** (2018)10 = (11111100010)2 = 11 bits = 1.375 bytes

(Since,1 byte = 8 bits)

**8.** Write CSE using ASCII code.Write CSE using Unicode.

**Ans 8:** CSE using ASCII: 67 83 69

CSE using Unicode: 0043 0053 0045

**9.** What were the shortcomings of Assembly languages?

**Ans 9:** The various shortcomings of Assembly languages are as follows.

* It takes a lot of time as well as effort to code for the same.
* It is complex and somewhat difficult to understand the code. For complex problems, programmer had to do the hard work of translating the mathematical problem into code. It did not make his work much easier.
* It occupies more memory of the computer as it contains long programs.
* It is not portable as it varies from architecture to architecture, as every computer hardware architecture consisted of its own machine language instruction set. Even if we write code in one machine, we had to write it again on another machine.

**10.** What were the advantages and limitations of ALGOL?

**Ans 10:** The variousadvantages of ALGOL:

* ALGOL introduced the idea of data-types.
* ALGOL encouraged the production of well-structured programs. It had introduced many new features such as the array structure, control statements and supported procedures and recursions. It became the first language to apply nested function definitions with lexical scope.
* It also became the first language to receive a standardized notation and a standard or formal definition.
* By becoming architecturally neutral, it also achieved machine independence which could be run on any machine.

The various limations of ALGOL:

* Larger programs were difficult to understand and debug.
* It had features that were too flexible, making it tough to understand and inefficient to implement.
* The instructions were being processed by a single processor. This model could not work efficiently for multiple CPUs being executed parallelly.

**11.** What is the difference between parameters and arguments of a procedure? Explain with appropriate examples.

**Ans 11:** A parameter represents a value that is used when we define a method and these parameters can change with different procedure calls. An argument, on the other hand is the actual value we pass in the parameter when calling the method.

For example:

*procedure findarea(l, b: in integer)*

*x, y, res: integer;*

*begin*

*x :=l;*

*y :=b;*

*res:=l\*b;*

*print, res, ‘is our calculated area.’*

*end findarea;*

Here, in the above procedure l, b are parameters.

Calling this procedure: We are now passing the arguments, where 5 and 10 are arguments.

findarea(5,10)

Thus, arguments are the actual values we pass while calling the method.

**12.** Write a recursive factorial function/method in C++ or Java (No limitation on number of lines).

**Ans12:** We can use JAVA to write a recursive factorial method.

*public static long factorial(long x)*

*{*

*if(x==0)*

*{*

*return 1;*

*}*

*else*

*{*

*if(x==1)*

*{*

*return x;*

*}*

*long result= x \* factorial(x-1);*

*return result;*

*}*

*}*

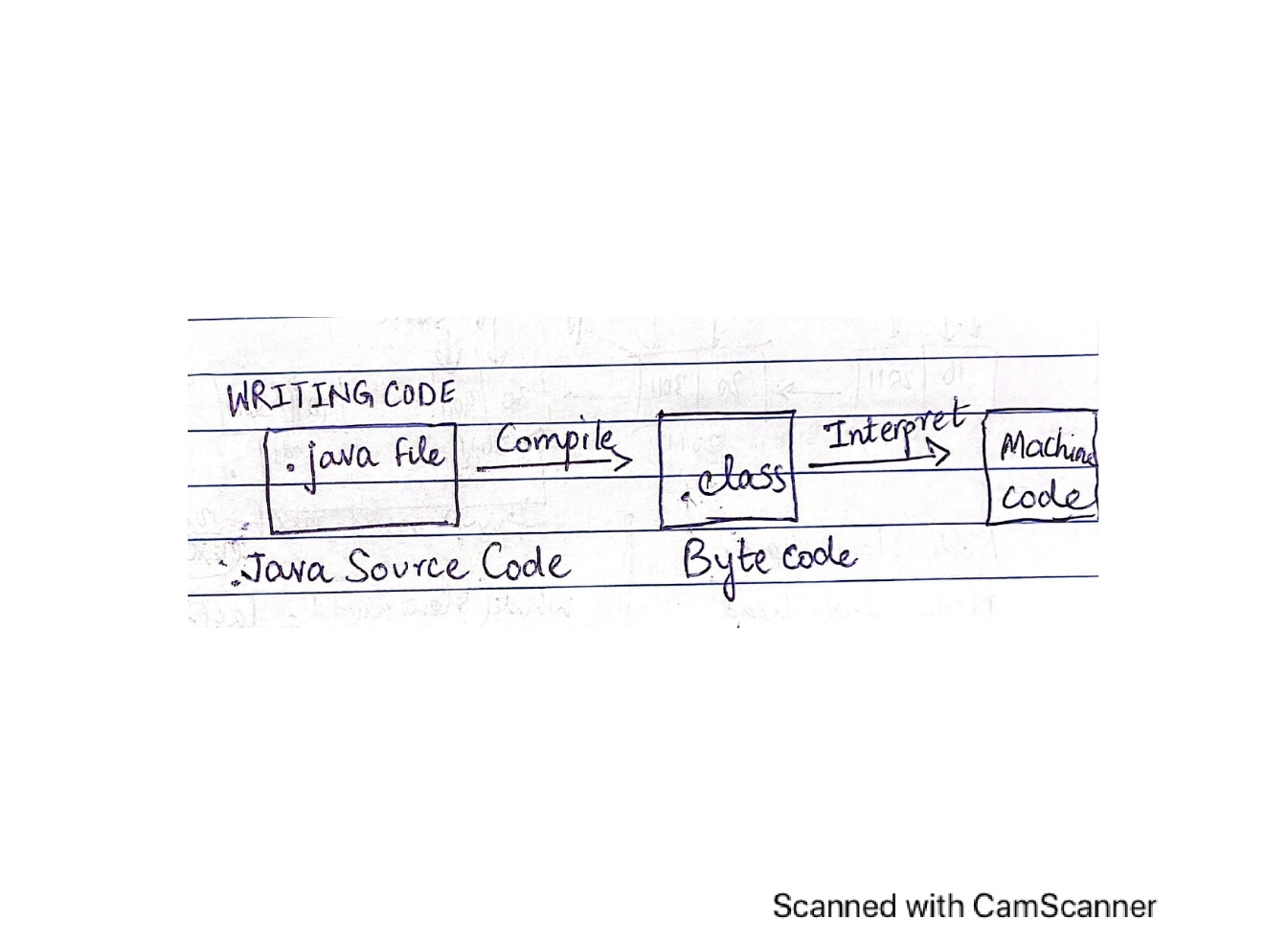
**13.** Explain Language Syntax and Language Semantics in your own words.

**Ans13:** Language syntax is the way/rules of writing a statement. It defines certain rules and guidelines to write statements, much like the grammar of natural languages.

Language semantic is something that describes what the meaning of a sequence of statements of code actually is. Language semantics check whether some sense is actually being generated from the code.

**14.** Explain how Java codes are compiled and then interpreted.

**Ans 14:** Java does both compilation and interpretation. The source code written in JAVA is first compiled by the JAVA compiler which converts the code into the bytecode. This bytecode is platform independent and saved with the file extension .class and then it goes to JAVA runtime environment, containing the JAVA Virtual Machine (JVM). JVM then interprets and translates bytecode into native machine code.



**15.** Explain the following terms:

a. Syntactic sugar

b. API

**Ans 15: a. Syntactic sugar -** Syntactic sugar are the shorthand notations or shortcuts provided by a programming language to make it easier to read and write. This is done to simplify a programmer’s tasks. For example:

In C++ or Java, x=x+10; can also be written as x+=10;

**b. API -** API is the abbreviated form of Application Programming Interface. API allows two applications or computer programs to interact with each other. For example, we use many Google APIs which allow communication with Google services like, Google Maps.

**Extra credit (bonus question):**

**16.** Do you think that C programming language is a successful programming language even though it’s not the most popular language today? Explain your reasonings.

**Ans 16:** Yes, C programming language is a successful language despite not being popular in the modern day. In fact, it is currently indispensable for some kinds of low-level systems programming,

* Though old, C language has been around for a pretty long time and still in use. Many programmers have attained their skills by learning C.
* C language is pretty simple and easy to understand to start learning about programming. One does not need to be an expert to get started. It is easy to read/ write codes in C.
* It also has in-built library functions for developing a program.
* Speed of compilation is very high. Many modern-day CPUs are built on C.
* Many other programming languages like C++, JAVA are based on C.
* Code written in C can be directly compiled into machine language.