Interpreter

Group E

Member Names:

Tousif Irshad	2472
Shahid Ashraf	2485
Muhammad Shoaib	2466
Ibrar Shami	2494
Salman Raza	2473
Jawad	2477
Mubashar Younus	2478
Muaz khan	2498

Interpreter

- All programming language instructions must be expressed in binary code before the computer can perform them.
- Computer languages can be grouped according to which translation process is used to convert the instructions into binary code:
  - Assemblers
  - Interpreters
  - Compilers

Interpreter

Interpreter: An **interpreter** is a computer program that directly executes, i.e. performs, instructions written in a programming or scripting language **or**

An interpreter translates high-level instructions into an intermediate form, which it then executes.

Translates one line of the program into binary code at a time:

Interpreting can be done in different ways, depending on the setting, how much time is available, how many languages are used, and whether or not technology is available.

Interpreter

Interpreted languages have become increasingly popular due to demands for rapid program development

Interpreter Descriptions:

Categories of interpreter types:

* Three interpreters types of working and performance.

* General types of Interpreter

Interpreter

Three interpreters types of working and performance.

- *MIPSI (Million instruction in per second)*
- *Perl*
- *Tcl (Tool command language)*

Interpreter

MIPSI:-

- An instruction-level emulator that executes
- multithreaded processors
- as a teaching tool in architecture and operating system classes
- initial stages of a CPU the fetch, decode and execute stages performed
  explicitly in software

Interpreter

Perl:-

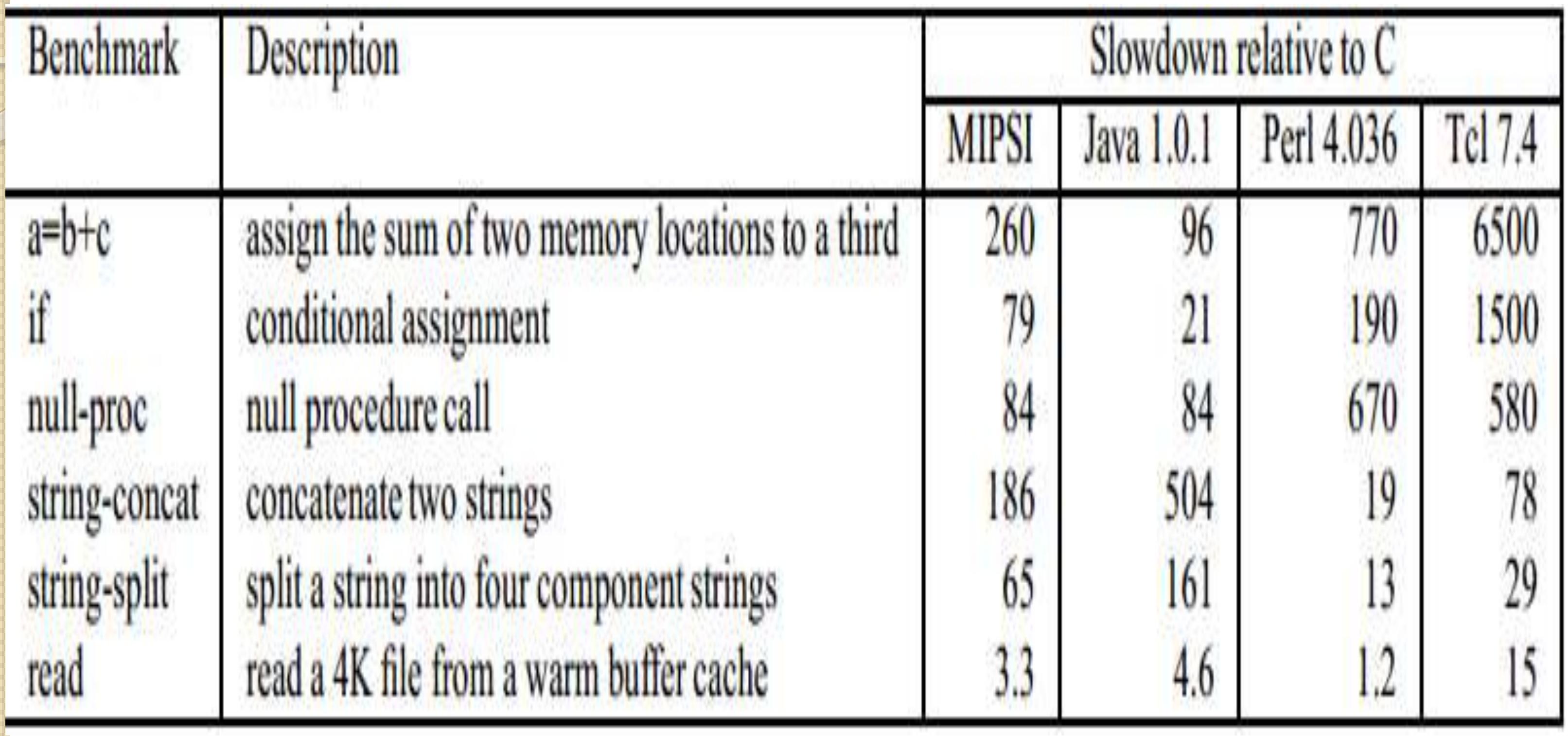
- *Perl* is a scripting language designed for manipulating text, data, files, and processes. Perl supports a variety of advanced programming abstractions useful for string and file processing, including regular expressions a high-level I/O interface, automatic memory management, and associative arrays. Perl programs are not interpreted directly

Interpreter

TCl:-

- *Tcl* is an embeddable command language that allows applications to be customized. Tcl is also used as a stand-alone programming environment, providing a programming language interface that includes basic functionality comparable to that found in Perl,
- One popular extension to Tcl is the Tk toolkit

Interpreter



Benchmark	Description	Slowdown relative to C			
		MIPSI	Java 1.0.1	Perl 4.036	Tcl 7.4
a=b+c	assign the sum of two memory locations to a third	260	96	770	6500
if	conditional assignment	79	21	190	1500
null-proc	null procedure call	84	84	670	580
string-concat	concatenate two strings	186	504	19	78
string-split	split a string into four component strings	65	161	13	29
read	read a 4K file from a warm buffer cache	3.3	4.6	1.2	15

This table shows the slowdown of each microbenchmark relative to the equivalent operation implemented in C

Working of Interpreter

- An instruction is **fetched** from the original source code.
- The Interpreter checks the single instruction for errors. (If an error is found, translation and execution ceases. Otherwise…)
- The instruction is translated into binary code.
- The binary coded instruction is **executed**.
- The fetch and execute process repeats for the entire program.

Interpreter

The steps to run a program via an interpreter:

1. Edit the Program
2. Debug or Run the Program
3. This is a far faster process and it helps novice programmers edit and test their code quicker than using a compiler. The disadvantage is that interpreted programs run much slower than compiled programs. As much as 5-10 times slower as every line of code has to be re-read, then re-processed.

Interpreter

An interpreter generally uses one of the following **Strategies for program execution**:

- parse the source code and perform its behavior directly
- translate source code into some efficient intermediate representation and immediately execute this
- explicitly execute stored pr-compiled code made by a compiler which is part of the interpreter system

Interpreter types

- There are two main types of interpretation:
  - Simultaneous Interpretation
  - Consecutive Interpretation

Interpreter types

- Simultaneous Interpretation

  The interpreter sits in a booth, listens to the speaker in one language through headphones, and immediately speaks their interpretation into a microphone in another language.

- Interpreters therefore take turns of **about 30 minutes**.

Interpreter types

- **Consecutive Interpretation**

  The interpreter repeats the message after the speaker has finished.
- Used for:
  - Small group conferences
  - Court cases
- Very long speeches may be broken up into parts

Interpreter

- Computer programs are compiled or interpreted. Languages like Assembly Language, C, C++, Fortran, Pascal were almost always compiled into machine code. Languages like Basic, VbScript and JavaScript were usually interpreted.
- However, BASIC and LISP are especially designed to be executed by an interpreter

Interpreter

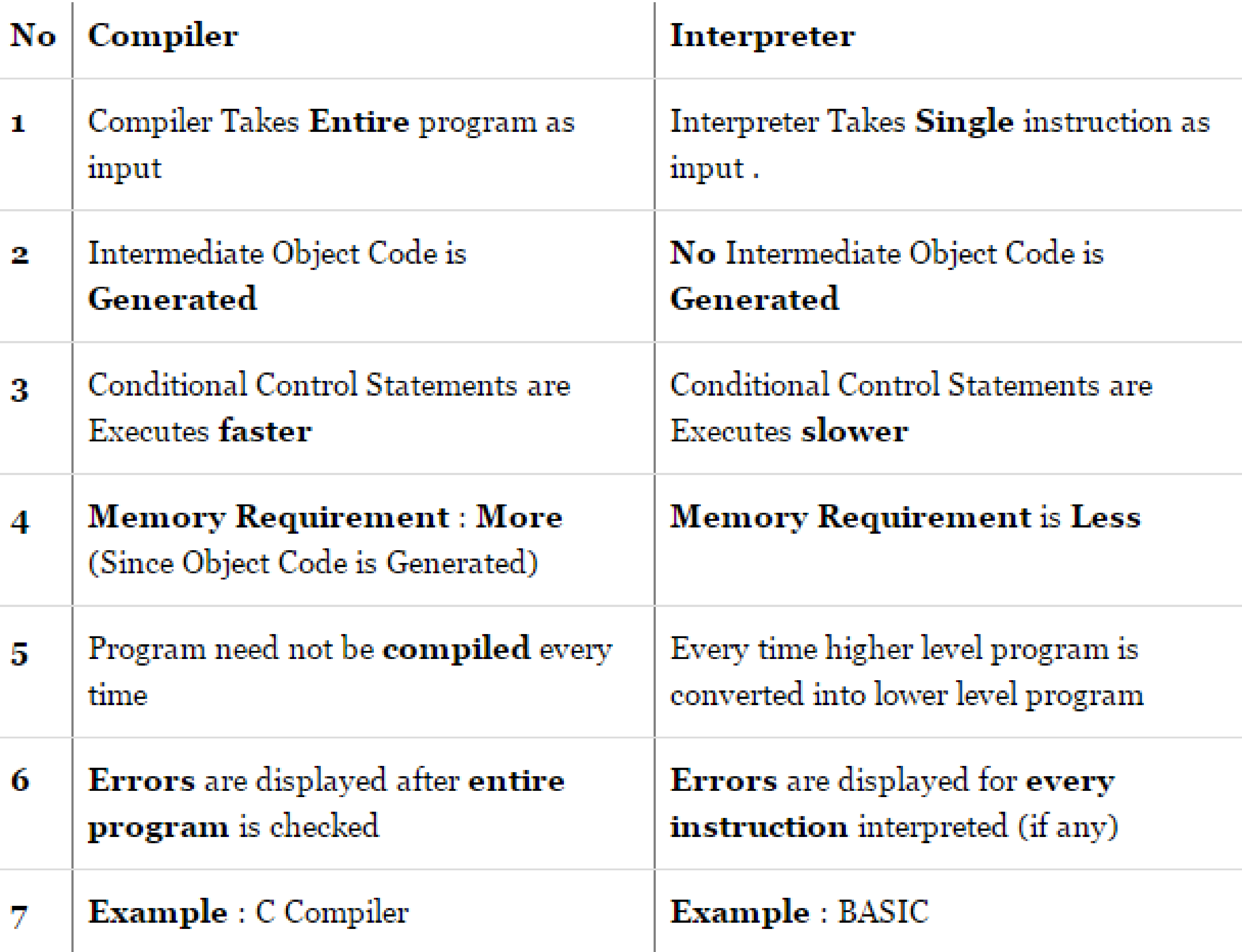
<u>Example</u>

- Using HTML to create Web Pages

  HTML (Hyper Text Markup Language): A computer language consisting of special codes intended to design the layout (or markup) of a Web page.

  - Web browsers interpret the HTML code and display the resulting Web pages.
  - Web browser: A program that displays information from the WWW.
  - Each line of HTML is called a tag (formatting instruction).

Difference between Compiler and interpreter



No	**Compiler**	**Interpreter**
1	Compiler Takes **Entire** program as input	Interpreter Takes **Single** instruction as input .
2	Intermediate Object Code is **Generated**	**No** Intermediate Object Code is **Generated**
3	Conditional Control Statements are Executes **faster**	Conditional Control Statements are Executes **slower**
4	**Memory Requirement** : **More** (Since Object Code is Generated)	**Memory Requirement** is **Less**
5	Program need not be **compiled** every time	Every time higher level program is converted into lower level program
6	**Errors** are displayed after **entire program** is checked	**Errors** are displayed for **every instruction** interpreted (if any)
7	**Example** : C Compiler	**Example** : BASIC