Hrishikesh.M

Assignment-1

PAGE No.
DATE / /

Q1 What is programming language?

→ Any notation for description of algorithms and datastructures called programming languages.

Q2 What are Types of Programming languages?

→ LLL – Low Level Language
HLL – High Level Language
MLL – Medium Level Language.

Q3 Diff between LLL, MLL, HLL

	LLL	MLL	HLL
1.	Speed is faster	Speed is medium	Speed is High.
2.	Consume less memory	Not high as compare to HL	They consume lot of memory
3.	They allow little of no abstraction	Significantly not more than LLL but small then HLL	they allow much more abstraction

4.	LLL are machine friendly	More friendly then LLL	HLL are human friendly
5. eg:	assembly language	eg C	eg C++ C#

~~Diff betw~~"

Why study programming languages?

⟶ * Become a better software engineer
1. Understand how to use Language
2. Appreciate implementation issues

* Better background for language selection
  familiar with range of language
* Understand issues / adv / disadv

~~Better background for Language selection~~
Better able to learn languages
* You might need to know a lot

Q4 Why study multiple programming language?

→ (1) Better understanding of implementation issues.
(2) Make easy to design new language.
(3) Ability to develop effective algorithms.
(4) Increase vocabulry of useful programming constructs.
(5) To allow better choice amongst alternative.
(6) To understand obscure
(7) To make good use of debuggers linkers & so on.

Q5 What are attributes of programming languages?

- Clearity, simplicity, unity
- Orthagonality - It refers to attribute of being able to combine various features of languages in all possible combination with every combination becomes meaningful
- Naturalness
- Support of abstraction
- Easy verification
- Cost of use.

PAGE No.
DATE / /

Q6 While designing languages which cost is considered?

→ While designing languages following cost are considered

cost for program execution
cost for program translation
cost for program creation
cost for program maintenance

Q7 Java compiler with 2 phases of compilation

→ Java compiler is a program that takes the text file work of a developer and compiles to platform independent Java file.

Two phases of compilation are:

1. Analysis phase.
2. Synthesis phase.

Q8 What is JVM?

→ Java virtual machine.

Q9 What is JIT?
→ Just in time compiler.

Q10 What is bytecode?
→ Machine independent intermediate output of java compiler. It can be interpreted as JIT compiler & converted to machine ~~learning~~ language.

Q11 Diff betw^n Compiler and Interpreter

Interpreter	Compiler.
① Translates programs one statement at a time	② Translates whole program at a time.
② No intermediate object code is generated	Generates intermediate object code.
③ Interpreter usually takes less amount of time to analyze the source code. eg Javascript, python.	③ Compiler usually take a large amount of time ~~is~~ to analyze the source code eg C, C++

PAGE No.
DATE / /

Q12 What is linker?
→ A linker is special prog that combines the object file generated by compiler or assembler.

Q13 What is loader?
→ The loader is special prog that ~~com~~ takes input of object code from linker loads it to main memory and ~~prepare special prog that combines the object file generated~~ this program for execution

Q14 Explain symbol table?
→ A symbol table is a data structure used by a language translation such as a compiler and interpreter.

What are types of compiler
1.] Single pass compiler
2.] two pass compiler
3.] Multipass Compiler.

Q15 Give any 4 tasks performed by the preprocessor

→ Include headerfile.
Simple textual representation
Macro expansion
Line control
Divide program into tokens
Remove white spaces & comments
Escape sequence.

Q16 Diff betwn C & C++

C	C++
(1) It follows top-down approach	(1) It follows bottom-up approach.
(2) The C compiler isn't able to execute the c++ code	(2) C++ is superset of C so C++ is able to run most of C code.
(3) It does not support encapsulation	(3) Support encapsulation
(4) Not good for securing	(4) Good for securing data.

Q17 Give any 4 tasks performed by preprocessor

→ (1) Include header file.
(2) Simple textual representation
(3) Macro expansion
(4) Line control
(5) Divide program in tokens
(6) Remove white spaces and comments
(7) Escape sequence

Q18 What is p-code?

→ It is portable code which also called pseudo code.
Rather than direct source code to machine code p-code creates 2 phases of translation.
1. Compilation
2. Interpretation or with second compiler such as JIT compiler.

Q19 Under which 2 circumstances bootstrapping is necessary?

→ 1. P-code version of compiler smaller than machine language this saves memory and disk space which is important

PAGE No.
DATE / /

2. They can run faster even more than of p-code of interpreter.