Questions on principle of programming language

Q1. What is principle of programming language?
Ans. It is a set of rules governed to communicate instructions to a machine, particularly a computer.

Q2. What are objectives of principles of programming language?
Ans. Objectives are-
• To introduce several different paradigms of programming
• To gain experience with these paradigms by using example programming languages
• To understand concepts of syntax, translation, abstraction, and implementation

Q3. What are the Paradigms of Programming?
Ans. Several paradigms are-
• Procedural
  – examples: C, Pascal, Basic, Fortran
• Functional
  – examples: Lisp, ML
• Object-oriented
  – examples: C++, Java, Smalltalk
• Rule-based (or Logic)
  – example: Prolog

Q4. Why there is need of so many paradigms?
Ans. The choice of paradigm and therefore language depends on how human’s best think about the problem other considerations are-
• Efficiency
• Compatibility with existing code
• Availability of translators

Q5. List the models of computation of language.
Ans. Models are-
• RAM machine
  – Procedural
• Directed acyclic graphs
  – Smalltalk model of O-O
• Partial recursive functions
  – Lisp and ML
• Markov algorithms
  – Prolog is loosely based on these
Q6. List various type of languages.
Ans. Various types of languages are:
• Document languages, e.g. LaTeX, Postscript
• Command languages, e.g. bash, MATLAB
• Markup languages, e.g. HTML and XML
• Specification languages, e.g. UML

Q7. What are the issues for languages?
Ans. Issues are:
• Can it be understood by people and processed by machines?
  – Although translation may be required.
• Sufficient expressive power?
  – Can we say what needs to be said, at an appropriate level of abstraction?

Q8. What is translation?
Ans. Translation is communication of converting the source code into target code.

Q9. What are different types of translation and their roles?
Ans. Types of translation are:
• Compilation
  – Translate instructions into suitable (lower level) machine code
  – During execution, machine maintains program state information
• Interpretation
  – May involve some translation
  – Interpreter maintains program state

Q10. What is trade’s off of translation.
Ans. Trade’s off of translation are:
• Compilation
  – lower level machine may be faster, so programs run faster
  – compilation can be expensive
  – examples: C
• Interpretation
  – more ability to perform diagnostics (or changes) at run-time
  – examples: Basic, UNIX shells, Lisp