

## Lecture 1: Introduction

1. Why is it useful for a programmer to have the ability to learn new languages, even though he or she may have a good knowledge of a number of programming languages?
2. Why is it essential to choose an appropriate programming language for a specific software solution?
3. Which programming language for scientific applications was the first to be used successfully?
4. Which is the first successful high-level programming language for business?
5. Which is the most popular markup language for Web development?
6. What is type checking? Give an example of how the failure to type check, at either compile time or run time, can lead to countless program errors?
7. How does the overall simplicity of a programming language affect its readability?
8. Explain how “writability” is used as a measure of how easily a language can be used to create programs?

## Lecture 2: language Translation

1. What is Von Neumann Architecture? Draw its architecture?
2. Define how the Von Neumann Architecture inspired the Imperative languages?
3. How many Types of Language Families? Define in brief each of them?
4. What are the implementation characteristics of compilation?
5. Define seven compilation phases processes?
6. What are the implementation characteristics of Hybrid Implementation?
7. What is Just-in-Time Compilation?
8. What is the Preprocessor? What is their disadvantage?

## Lecture 3: Names, Binding, Type checking , and Scopes

1. What is a reserved word?
2. What is the address of a variable?
3. What is type inference? Give an example.
4. What is deallocation of a memory cell?
5. After language design and implementation, what are the four times bindings can take place in a program?
6. What is the lifetime of a variable?
7. What is the use of a stack-dynamic variable?
8. What are the advantages and disadvantages of dynamic type binding?
10. Define static, stack-dynamic, explicit heap-dynamic, and implicit heap-dynamic variables? What are their advantages and disadvantages?

## **Lecture 4: Data Types**

1. What is a data type?
2. What are the different representations for floating-point values supported by Java?
3. What is an abstract data type?
4. Describe the different uses of the type system of a programming language?
5. Define descriptor and object.
6. What are the two most common structured data types in the imperative languages?
7. What mechanism is used to store negative integers in a computer?
8. What are the four signed integers supported by Java?
9. Define static, fixed stack-dynamic, fixed heap-dynamic, and heap-dynamic arrays. What are the advantages of each?
10. What languages support complex data type?

## **Lecture 5-A: Subprograms**

1. What are the three general characteristics of subprograms?
2. What is a subprogram call?
3. What is a subprogram definition?
4. What characteristic of Python subprograms sets them apart from those of other languages?
6. Describe the ways that aliases can occur with pass-by-reference parameters?
7. What are function declarations called in C and C++? Where the declarations often are placed?
8. Name one pure functional programming language that does not have mutable data?
9. What are positional parameters? Give an example of a language that allows positional parameters in addition to keyword parameters?
10. What is the use of a default value in a formal parameter?

**Good Luck**  
**Dr. Aghabi Nabil Abosaif**