Part I (30 pts) Introduction

Choose a high-level programming language (imperative, object-oriented, or imperative based O-O) that is not C/C++ or Java and write (using a double-line, 12pt font) your answers to the following questions:

1. What is the type of programming language (imperative, object-oriented, imperative based O-O, functional, or logic based) and when it was created?
2. Is its syntax related to the syntax of another known programming language?
3. Why was this language introduced? (What is/are the programming domain(s)?)
4. What is its implementation method(s)? (compiler implementation, pure interpretation, hybrid implementation (JIT compiler?)
5. Does the language have a runtime environment? If yes, how is it called? And where do you get it (web site URL)?
6. Are there any programming framework for this language? If yes, where and how do you get them (web site URL)?
7. What is (are) the program development environment(s) available for this language? And what is provided in each program development environment?
8. Where and how do you get a program development environment for this language?
9. Are there any IDEs that support this language? If yes, where and how do you get it?
Part 2 (110 pts) Basic Elements of the Programming Language

A. Comments (5 pts)
1. Rule for writing valid comments
2. Examples of comments

B. Identifiers (10 pts)
1. Rules for writing valid identifiers
2. Length of an identifier
3. Case sensitivity: are uppercase and lowercase characters different?
4. Connectors
5. Examples
6. Keywords/Reserved words: Are keywords reserved words?
7. Standard identifiers: does the language has standard identifiers?

C. Constant Data/ Literal Constants (20 pts)
- Possible constant data follow:
  - Boolean constants
  - Character constants
  - Integer constants
  - Decimal constants
  - Floating-point constants
  - String constants
  - Special constants

D. Variables, Basic Data Types and Type Binding (20 pts)
1. Name: how is it referred to?
2. Address: is the address of a variable accessible? If Yes, how?
3. Is the size of a data type dependent of the computer?
4. Possible Basic Data Types:
  - Signed integers
  - Unsigned integers
  - Floating-point Types
- Decimal
- Boolean types
- Character Types
- Character String Type (is there a basic data type to represent strings?)
- String Length
  o Static length strings, limited dynamic length strings or dynamic length strings?

5. Type Binding:
   - Is the type binding static or dynamic? Do we have type inference?
   - If the type binding is static, do we have implicit or explicit declarations of variables?

E. Arithmetic Expressions (15 pts)
   1. (character/integer/floating point) arithmetic operators?
   2. Rule for writing (character/integer/floating point) arithmetic expressions.
   3. Rule for evaluating arithmetic expression
   4. Does the language allow mixed-mode expressions? If yes, what is the type conversion rule?
   5. Order of operands evaluation.
   6. Are there restrictions on operand evaluation side effects?
   7. Does the language allow user-defined operator overloading?
   8. Does the language have decimal arithmetic operations?

F. Logical Expressions (10 pts)
   1. Relational operators?
   2. Logical operators?
   3. Precedence of basic operators?

G. Assignment Statements (15 pts)
   1. Syntax
   2. Are mixed-mode assignment allowed? If yes, what is the coercion rule?
   3. Is the assignment an operator?
   4. Do we have compound assignments?
   5. Initial values of variables?
   6. Naming constants?
   7. Increment and decrement operators?

H. Representations of String Data (15 pts)
   1. If strings are not represented by a basic data type, how are they represented?
   2. What are the operations on strings?
   3. Converting strings of digits to numerical values?
   4. Converting numerical values to strings?
Part 3 (55 pts) Executable Statements

A. Standard input. (5 pts)
B. Standard output. (5 pts)
C. Formatted output. (10 pts)
D. Implementations of Control structures:
   1. Two way selection. (5 pts)
   2. One way selection. (5 pts)
   3. Counter –controlled iteration. (5 pts)
   4. Logically controlled iteration. (5 pts)
   5. Multiple way selections. (10 pts)
E. Other forms of statement?

Part 4 (20 pts)

A. Structure of a program. (5 pts)
B. How to write and execute a program? (5 pts)
C. Write and execute the “Hello World!” program. (10 pts)

Part 5 (55 pts) Procedures and Functions

1. Types of procedures and functions. (5 pts)
2. Return types of functions. (5 pts)
3. General syntax of procedures and functions. (5 pts)
4. Types of parameters. (5 pts)
5. Defining and calling procedures and functions. (20 pts)
6. Local/global variables and other types of variables. (10 pts)
7. Procedure/function (name) overloading. (5 pts)

Part 6 (70 pts).

A. One-Dimensional Arrays
   1. Defining and initializing one-dimensional arrays. (10 pts)
   2. Processing one-dimensional arrays. (10 pts)
   3. One dimensional arrays as parameters of procedures/functions. (10 pts)
   4. One-dimensional arrays as return types of functions. (5 pts)
B. Two-Dimensional Arrays
   1. Defining and initializing two-dimensional arrays. (10 pts)
   2. Processing two-dimensional arrays. (10 pts)
   3. Two-dimensional arrays as parameters of procedures/functions. (10 pts)
   4. Two-dimensional arrays as return types of functions. (5 pts)

Part 7 Object-Oriented Programming