4143 - PLC
Fall 2023
Exam Two
November 21, 2023

Name: ____________________

1. Multiple Choice

1. Why are programming languages essential in technology and computer science?
   - A. They provide entertainment through games.
   - B. They serve as decoration for computer systems.
   - **C. They enable effective communication with computers and problem-solving.**
   - D. They are only necessary for web development.
2. Which programming languages are crucial for data analysis and machine learning?
   - A. HTML and CSS
   - **B. Python and R**
   - C. Java and C#
   - D. Swift and Kotlin
3. What is the role of programming languages in web development?
   - A. They enhance the performance of hardware.
   - **B. They are used to build diverse webpages and applications.**
   - C. They are mainly used for creating desktop applications.
   - D. They are not used in web development.
4. What drives innovation in the field of programming languages?
   - **A. The development of new languages for specific needs or emerging technologies.**
   - B. The maintenance of old programming languages.
   - C. The decrease in the use of technology.
   - D. The focus on a single programming language.
5. What opportunities arise from proficiency in programming languages?
   - A. Decreased demand in the job market.
   - B. Limited career paths.
   - **C. High job security and competitive salaries.**
   - D. None of the above.

6. Which of the following is an example of a syntax error?

   A. Using a variable that has not been declared.

   B. Trying to divide by zero.

   C. Missing a semicolon at the end of a statement.

   D. All of the above.

7. What would be the result of a semantic error in your code?

   A. The code will not compile.

   B. The code compiles but does not do what it is intended to do.

   C. The code will become more optimized.

   D. The code will run faster.

8. In Go, to access a module's function, the ____________ operator is used.

   A. The . (dot) operator is used.

   B. The **->** operator is used.

   C. The **[]** operator is used.

   D. The **{}**operator is used.

9. Which of the following is a statically typed and compiled language?

   A. Python

   B. Perl

   C. Go (Golang)

   D. JavaScript

10. In compiled languages, what is the role of a compiler?

    A. To execute code line by line

    B. To translate source code into machine code

    C. To interpret bytecode

    D. None of the above

11. Which language is known for its cross-platform portability due to the lack of a separate compilation step?

    A. C++

    B. Rust

    C. Python

    D. Go (Golang)

12. What is a major advantage of interpreted languages over compiled languages?

    A. Higher execution speed

    B. Ease of development and cross-platform compatibility

    C. More efficient memory usage

    D. All of the above

13. Which of the following optimizations might a compiler perform?
    - **A. Dead code elimination**
    - B. Just-In-Time compilation
    - C. Line by line interpretation
    - D. Dynamic typing

14. What is the execution method of interpreted languages?
    - A. Compilation to machine code before running
    - **B. Execution of code line by line**
    - C. Both A and B
    - D. None of the above

15. Which of the following virtual machines uses Just-In-Time compilation for performance?
    - A. Python Virtual Machine (PVM)
    - **B. Java Virtual Machine (JVM)**
    - C. Go (Golang)
    - D. None of the above

16. Semantic errors are errors in the code that:
    - A. Will not allow the code to compile.
    - **B. Allow it to compile but produce incorrect results.**
    - C. Can be Type-related errors that are caught during compilation.
    - D. Are lexical connotations that are ill formed and expose syntactic deficiency's

17. Which of the following statements is true about static typing?
    - A. Type checking is performed at runtime.
    - B. It allows implicit type changes.
    - **C. Type-related errors are caught during compilation.**
    - D. All variables are untyped until runtime.

18. In statically typed languages, what is required for variables?
    - **A. Explicit type declarations**
    - B. Variables can change types implicitly
    - C. No type declarations are needed
    - D. Type checking only at runtime

19. Which languages are examples of statically typed languages? (Choose two)
    - A. Python
    - B. JavaScript
    - **C. Java**
    - **D. C++**

20. What does dynamic typing prioritize over static typing?

    A. Early error detection

    B. Type safety

    C. Development flexibility and speed

    D. Explicit type declarations

21. What does a “strongly typed” language enforce?

    A. Implicit type conversions

    B. Flexible type usage without explicit conversions

    C. Strict type enforcement without mixing types without conversion

    D. None of the above

22. True or False: In a weakly typed language, you can add a string to an integer without explicitly converting one of them.

    A. True

    B. False

23. What is reflection in programming languages primarily used for?

    A. Optimizing code execution at compile time.

    B. Hardcoding the structure and behavior of a program.

    C. Examining and modifying a program’s structure and behavior at runtime.

    D. Increasing the compile-time type safety of a program.

24. What is the output of the following Python code?

```
class MyClass:
def __init__(self):
self.x = 10

setattr(MyClass, 'y', 20)
obj = MyClass()
print(obj.y)
```

    A. AttributeError

    B. 10

    C. None

    D. 20

25. In Python, how can you get a list of all function objects in a module?

    A. Using the `dir()` function on the module.

    B. By iterating over the module’s `__dict__` attribute.

    C. By manually inspecting the module’s source code.

    D. Using the `inspect.getmembers()` function with `inspect.isfunction()` .

26. What is the purpose of the `__doc__` attribute in Python classes and functions?

    A. To set the name of the class or function.

    B. To optimize the runtime performance.

    C. To store the docstring of the class or function.

    D. To declare private member variables.

27. Consider the following Python code snippet:

```
total = 0
for i in range(5):
quad total += i
```

In terms of programming paradigms, how would you classify this code?

    A. Procedural programming because it uses a for-loop.

    B. Imperative programming because it sequentially changes the program state.

    C. Object-oriented programming because it uses Python's built-in range function.

    D. Functional programming because it calculates a sum.

28. What are attributes in Python?

    A. Functions that are applied to class methods.

    B. External libraries for type checking.

    C. Values associated with an object, such as data or methods.

    D. Metadata for optimizing Python code execution.

29. Which of the following is a correct example of a function with annotated parameters in Python?

A.

```
def greet(name):
return 'Hello ' + name
```

B.

```
def greet(name: 'name should be a string'):
return 'Hello ' + name
```

C.

```
def greet(name: str) -> str:
return 'Hello ' + name
```

D.

```
def greet(name) -> str:
return 'Hello ' + name: str
```

30. What are annotations in Python primarily used for?

   A. Changing the way a function executes.

   B. Providing metadata such as type hints to tools and libraries.

   C. Storing values that an object will use during execution.

   D. Decorating functions to add additional functionality.

31. What is the purpose of a decorator in Python?

   A. To modify or enhance the behavior of a function or method.

   B. To provide a strict type system for Python variables.

   C. To annotate the parameters of any function with metadata.

   D. To optimize Python code for better performance.

32. Which of the following code samples is an example of procedural programming?

A.

```
x = 10
y = 20
sum = x + y
print(sum)
```

B.

```
def calculate_sum(x, y):
return x + y
print(calculate_sum(10, 20))
```

   C. Both A and B

   D. Neither A nor B

Fill In the Blanks

Word Choices

Annotations	Attributes	Byte Code	C++	Compile Time
Decorators	Development Speed	Dynamic Typing	Encapsulation	Go
Imperative	Inheritance	Lexical	OOP	Performance
Perl	Portability	Procedural	Properties	Python
Reflection	Run Time	Semantics	Static Typing	Strongly Typed
Syntax	Type-Safe	Virtual Machine	Weakly Typed	

Use the words above to assist you into entering the correct response for the questions below. Of course there are more than necessary. But all the correct answers are in the table.

33. Compiled languages like A)**Go (Golang)** and B)**C++** are preferred when developing applications that need to have C)**Performance** rather than ease of development.

34. Interpreted languages such as A)**Python** & B)**Perl** are often used for their C)**Portability** and D)**Development** E)**Speed** which is a direct contrast to compiled languages.

35. Dynamic typing performs type checking during A)**Run Time**, whereas static typing performs it during B)**Compile Time**.

36. In programming, A)**Syntax** defines how you write the code, whereas B)**Semantics** is about the meaning behind the code you write.

37. The operation of adding two integers together in programming is defined by the A)**Semantics** of the language.

38. Forgetting a semicolon when writing a C++ solution would violate the A)**Syntax** of the language.

39. Values associated with an object, such as data or functions, that are like variables for an object are called A)**Attributes**.

Matching

Language Features

40. **Match the features to the correct programming language descriptions below:**

Letter	Goal	Letter	Goal	Letter	Goal	Letter	Goal
A	Readability	F	Expressiveness	K	Performance	P	Maintainability
B	Efficiency	G	Safety	L	Paradigm Support	Q	Security
C	Portability	H	Community	M	Standardization	R	Ecosystem
D	Scalability	I	Interoperability	N	Ease of Learning	-	-
E	Abstraction	J	Conciseness	O	Specialization	-	-

#	Description	Answer
a.	Code can run on multiple platforms, often through VMs or interpreters.	C
b.	Allows developers to write code with maximum control over system resources.	K
c.	Tailored to specific programming paradigms like OOP or functional programming.	L
d.	Prioritizes a syntax and structure that make code easy to understand and maintain.	A
e.	Supports development from small scripts to large systems, enabling growth.	D
f.	Focused on reducing verbosity and promoting clear developer intentions.	F
g.	Aims to prevent common errors and protect against vulnerabilities.	G
h.	Ensures that a language has a wide range of libraries and an active developer base.	H
i.	Designed for seamless operation with other programming languages.	I
j.	Aims for succinct code that reduces errors and is easy to comprehend.	J
k.	Ensures consistency and avoids fragmentation across different platforms.	M
l.	Simple syntax and a gentle learning curve for new programmers.	N
m.	Created for specific tasks, like database management or scientific calculations.	O

GoLang Goals

A	Simplicity and Clarity	B	Efficiency	C	Concurrent / Parallel Programming
D	Safety	E	Robust Standard Library	F	Strong Typing and Static Compilation
G	Built-in Concurrency Primitives	H	Tooling	I	Cross-Platform Compatibility
J	Garbage Collection	K	Orthogonality	L	Open Source and Community-Driven

41. **Match the following descriptions with the Golang goals from the table above:**

#	**Description**	**Answer**
a.	Focuses on clean and minimalistic syntax for better readability.	**A**
b.	Targets quick compilation and fast runtime performance.	**B**
c.	Facilitates writing scalable and efficient software that runs concurrently.	**C**
d.	Prioritizes the reduction of common programming errors through features like garbage collection.	**D**
e.	Provides extensive built-in functionalities for a wide range of programming needs.	**E**
f.	Catches errors at compile time with a reliable type system.	**F**
g.	Embeds concurrency models in the language itself for easier parallel programming.	**G**
h.	Includes a complete set of tools for code formatting, management, and testing.	**H**
i.	Allows code to be compiled on different platforms without changes.	**I**
j.	Automates memory management to ease the development process.	**J**
k.	Promotes code with separated concerns for maintainability.	**K**
l.	Benefits from community contributions and an open-source ecosystem.	**L**

Language Types

42. **Match the following languages with their type:**
A = Compiled
B = Interpreted
C = Combination

	Language	Type
a.	C#	**C**
b.	Python	**B**
c.	C++	**A**
d.	Perl	**B**
e.	Javascript	**B**
f.	Java	**C**
g.	Go (Golang)	**A**

43. **Match the following language characteristics with their respective typing discipline.**

S = Static Typing
D = Dynamic Typing

	Characteristic	Discipline
a.	Implicit type conversion	**D**
b.	Requires explicit type declarations	**S**
c.	Prioritizes early error detection	**S**
d.	Allows variables to change types implicitly	**D**
e.	Prioritizes development flexibility	**D**

Short Answer

44. Explain in your own verbiage what the following four words mean and how they relate. Give language names and code examples to bolster your discussion. The words are: **Dynamic**, **Static**, **Weak**, **Strong**.

Solution:

Dynamic and Static Typing:

Dynamic typing refers to the flexibility of a programming language where the data type of a variable is determined at runtime. In dynamically typed languages like Python and JavaScript, you can change the type of a variable during execution. For instance, you can assign an integer value to a variable and later assign a string to the same variable.

In contrast, **static typing** means that the data type of a variable is defined at compile-time. Languages like Java and C++ fall into this category. Once you declare a variable with a specific type, you can't change it. It's checked at compile-time for compatibility, ensuring type safety and reducing runtime errors.

Weak and Strong Typing:

Weak typing implies that a programming language allows for implicit type conversion. In weakly typed languages like JavaScript, you can perform operations between different data types without explicit type casting. This can lead to unexpected behaviors and potential errors.

On the other hand, **strong typing** enforces strict type checking, preventing operations between incompatible data types. Languages like Python and Ruby are strongly typed, and they require explicit type conversions to perform operations on variables with different data types.

The Relationships:

- **Dynamic and Weak Typing:** Languages like JavaScript combine dynamic and weak typing. You can change the data type of a variable dynamically, and implicit type conversion is allowed.

- **Dynamic and Strong Typing:** Python is an example of a language that combines dynamic and strong typing. You can change the type of a variable dynamically, but type conversions must be explicit.

- **Static and Strong Typing:** Java exemplifies static and strong typing. The data type of a variable is determined at compile-time, and type conversions are strict and explicit.

- **Static and Weak Typing:** This combination is relatively rare in mainstream languages because it can lead to unexpected behaviors and errors. It's often considered less desirable for software development.

Note: While C and C++ are generally considered to have weak or permissive type systems in certain contexts, it's essential to clarify that these languages also allow you to write strict and type-safe code by being explicit with type conversions and adhering to best practices. The permissive nature of C and C++ can be a double-edged sword, offering flexibility and control but also requiring careful attention to prevent programming errors and unexpected behavior.

45. Give a decent explanation of something you studied but wasn't on the exam. I cannot promise lots of points, but, if it is a good explanation it can really help you.

Solution:

I took anything that was exam related, or very close.