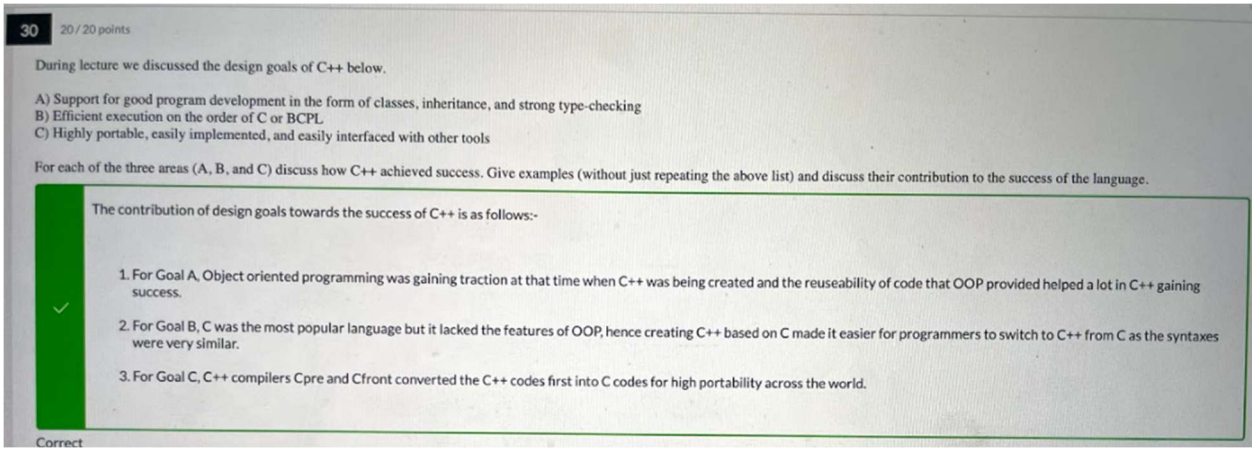
1. A language is considered to be _____ if you can add features to it.
    a. Extensible
2. You can only create a new language after you write a formal specification
    a. False
3. There are no new languages being created
    a. False
4. There is no generally accepted formal method for describing semantics
    a. True
5. A notation for communicating to a computer what we want to do
    a. Programming language
6. A language lacks this when dissimilar things look or act similarly
    a. Uniformity
7. An interpreter executes a program directly
    a. True
8. In the C language, functions are defined the same way that math defines functions
    a. False
9. In class we discussed the design goals of various languages. This question focusses on Java. Discuss (3) aspects of Java that contributed to its success and explain how they contributed to the language's adaptation across so many platforms
    a. It was based on C/C++ which makes its code easy to read/write/understand
    b. It is highly secure as it eliminated pointers, it promoted oop which at the time was getting tractions
    c. It is highly portable since it uses Java VM and making its bytecode acceptable on every platform.
    d. Additionally, upon it's inception web developments was a new concept and the programmers needed to run their programs on clients machines regardless of the platform where java came in handy
10. The reason a language defines virtual machines (like the Java VM) is to target a single physical CPU.
    a. False
11. There is no difference between a complier and an interpreter
    a. False
12. Writability is the primary factor in determining whether a language is successful
    a. False
13. Some languages needed better hardware performance before they become practical to use
    a. True
14. Both complier and an interpreter translate programs
    a. True
15. Abstraction is a technique to manage complexity
    a. Ture
16. Uniformity mean a design in which similar things look and act similar
    a. True
17. A language is considered to be _____ if it prevents a programmer from compiling or executing any statements or expressions that violate the definition of the language.
    a. Semantically safe
18. It is possible to focus too much on a particular design goal or goals
    a. True
19. A language that does not have a defined standard is immune from problems of ambiguity

    a. False
20. Ambiguity in a programming language is desired feature
    a. False
21. A language that is extremely expressive is also always extremely reliable
    a. False
22. Language paradigms are a form of syntactic sugar
    a. False
23. Languages that satisfy the criterion of regularity are said to adhere to the principle of least astonishment
    a. True
24. Backwards compatibility with another language may be valid cause of irregularity
    a. True
25. Translates an entire program into executable code and does not execute it
    a. Compiler
26. (2parts) 1 – What is the difference between programmer efficiency and runtime efficiency?
    2 – Which is more important? Explain your answer
    a. 1 – Programmer efficiency - The ease of a programmer to read, write and understand the complexities of a program code
       Runtime efficiency - A program can take seconds/minutes/hours/ years to execute, the time taken by a program to execute is known as runtime efficiency
    b. Run time is more important
        i. Its better for productivity
27. The output of an interpreter is something that can be executed later
    a. False
28. When designing a new language there is no benefit in starting from an existing language
    a. False
29. Python is not suitable for large or time-critical systems
    a. True
30. C++
    a. 

31. Match the vocabulary
    a. Assembler – A program that translate the symbolic assembly language code to binary machine code
    b. Machine language – The binary form of instructions for a CPU
    c. Assembly Opcodes – Basic control abstraction
    d. Syntax – The structure of a language

    e. Abstraction – Technique to manage complexity
    f. Variable – Basic data abstraction
    g. Array – Structured data abstraction
    h. Thread – unit control abstraction
    i. Semantics – The meaning of a language
    j. Branch – Structured control abstraction
    k. Assembly language – Mnemonic symbols to represent binary machine codes
    l. Interface – Unit data abstraction
32. If a language is _____, its constructs do not behave differently in different contexts
    a. Orthogonal
33. Using a virtual machine as part of a language design provides better hardware performance
    a. False
34. Matching
    a. Uniformity – Similar things look and act similar
    b. Regularity – Refers to how well the features of a language are integrated
    c. Extensibility – Allows new features to be added
    d. Orthogonality – Constructs do not behave differently in different contexts
    e. Orthogonality – Refers to the consistency of appearance and behavior of language constructs
    f. Generality - Avoids special cases
35. Certified as the first high-level language
    a. FORTRAN