Quiz: 20 questions (5 points each)

1. Why is String capitalized?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A. It’s a primitive type. | B. It’s an object. | C. It’s static. | D. It’s a library. |

1. Which is **not** a primitive type?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A. int (integer) | B. float (floating point number) | C. Person (name, age, etc.) | D. char (character) |

1. Which is: int a; ?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A. declaration | B. assignment | C. arithmetic operation | D. string operation |

1. Which is: name = "hello"; ?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A. declaration | B. assignment | C. arithmetic operation | D. string operation |

1. Which is: a= b/c; ?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A. declaration | B. assignment | C. arithmetic operation | D. string operation |

1. Which is: fullName = firstName+lastName; ?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A. declaration | B. assignment | C. arithmetic operation | D. string operation |

1. What must you do before you run a java program?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A. halt | B. compile | C. debug | D. loop |

1. What is the process of correcting mistakes?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A. halt | B. compile | C. debug | D. loop |

Fill in the following table with the correct symbols or key words.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | double quote |  |  |  | semi-colon |
|  |  | assignment operator |  |  |  | line comment |
|  |  | open parenthesis |  |  |  | is equal to? |
|  |  | close parenthesis |  |  |  | integer |
|  |  | open curly brace |  |  |  | single precision float |
|  |  | close curly brace |  |  |  | division |

Bonus: Write a line that would output the variable firstName and LastName with a space in between.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**byte code, comment, compile, executable, formal language, high-level language, low-level language, natural language, object code, problem-solving, portability, print statement, source code, statement**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** The process of formulating a problem, finding a solution, and expressing the solution.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** A programming language like **Java** that is designed to be easy for humans to read and write. **Scripting languages (python, basic) are also high-level.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** A programming language that is designed to be easy for a computer to run. Also called “machine language” or “assembly language.”

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** Any of the languages people have designed for **specific purposes**, like representing mathematical ideas or computer programs. All **(current)** programming languages are formal languages.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** Any of the languages **people speak** that have evolved naturally.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_: Can run on more than one OS.** A property of a program that can run on **more than one kind of computer**.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** To translate a program in a high-level language into a low-level language, all at once, in preparation for later execution. **Usually into machine code.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** A program in **a high-level language**, before being compiled.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** The output of the **Java compiler**, after translating the program.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** Another name for **object code that is ready to run**.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** A **special kind** of object code used for **Java programs**. Byte code is similar to **machine code**, but it is **portable**, like a high-level language.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_: One line of code or one command.** A part of a program that specifies a computation.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** A statement that causes **output** to be displayed on the screen.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** A part of a program that contains notes about the program, but that has no effect when the program runs.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** An **error** in a program.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** The **structure** of a program.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** The **meaning** of a program.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_: For the computer to read and find the “meaning” of your software.** To examine a program and analyze the syntactic structure.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** An error in a program that makes it **hard/impossible to parse** (and therefore impossible to compile).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** An error in a program that makes it fail at run-time. Also called a run-time error.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** An error in a program that makes it do something other than what the programmer intended.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:** The process of finding and removing any of the three kinds of errors.