Task 1.

Statistically Typed Languages are the programming languages that checking type at the compile

time, while Dynamically Typed Languages are the programming languages that checking type at the

run time.

Strongly Typed Languages are the programming languages that consider the type of data strictly

while Loosely Typed Languages are the programming languages that don’t consider the type of data

strictly.

Java is a statistically typed and strongly typed programming language

Task2.

Case sensitive – upper case and lower case letters in identifiers are treated as two different

identifiers.

Example

int variable = 42;

int Variable = 23;

System.out.println(variable);

System.out.println(Variable);

here, no errors and gives values separately.

Java is case sensitive

Case insensitive – it does not differentiate between uppercase and lowercase.

Example treat variable / Variable / VARIABLE as same.

SQL is case insensitive

Case Sensitive-Insensitive – mixed with case insensitive and case sensitive.

Example

treat myVariable and MyVariable as different and treat if and IF as same

JavaScript is Case Sensitive-Insensitive

Task3.

Identity Conversion is a type of type conversion that involves assigning a value to a variable of the

same type, or passing an argument to a method where the argument's type matches the parameter

type exactly.

Example

int number = 42;

int anotherNubmer = number;

long myLong = 151316;

long myNextLong = myLong;

here firstly, a value is assigned to number and that number is assigned to another variable with same

type.

Task4.

Primitive widening conversions are converting a variable with a lesser bits to a variable with a

higher bits. These happen implicitly.

Byte>short>int>long

float>double

Task 5.

run time constant- the value of the constant is known when the project runs

example

final int Const1 = 10\*Math.Random();

final int myInt3 = myByte; // RUN TIME CONSTANT

compile time constant- the value of the constant is known when the project compiles.

Example

final int Const2 =20;

final int myInt4 = 15; //COMPILE TIME CONSTANT

Task6.

Implicit conversions happens automatically and explicit conversions are not happening

automatically without external forcing (casting the target type at the end ).

conditions to met to happen an implicit narrowing primitive conversion

1. an assigning context

2. a compile time constant

3. within the bit range of target type

Task7.

The but structure if the float is different from long. Even the number of bits is lesser, float can store

larger range of values, but with a less accuracy.

Task8.

The choice of using int as the default data type for integer literals and double as the default data

type for floating-point literals in Java is primarily based on balancing practicality, efficiency, and

backward compatibility.

32 bit processor - optimized for 32 bits

64 bit processor - optimized for 32 bits / 64 bits

to maintain consistency between pr. languages selected as follows

int - 32 bits long - 64 bits —> selected int considering performance

float - 32 bits double - 64 bits —> selected long considering size

considering performance, byte or short was not considered as default data type as the range is smaller.

Task9

because they don’t lose information while narrowing

Task10.

When check in to the chart it have to go through a path of short>int>char which the first step is a

widening primitive conversion and the second step is a narrowing primitive conversion. (but

actually it happens directly as short>char)