Assignment 02

1. Question 01

- Statically-typed Programming Language

  A statically typed programming language is a language which does the type checking process in the compile time.

- Dynamically-typed Programming Language

  A dynamically typed programming language is a language which does the type checking process in the runtime.

- Strongly-typed Programming Language

  A strongly typed programming language is a language which considers the data types strictly

- Loosely-typed Programming Language

  In a loosely typed programming language, the data types are not considered strictly.

Java falls into both statically typed and dynamically typed programming languages.

Java is a strongly typed programming language.

2. Question 02

- Case Sensitive Programming Language

  In a case-sensitive language, the distinction between uppercase and lowercase letters matters when letters are compared. This means that 'A' is considered different from 'a,' and two words that differ only in their letter case are treated as distinct.

  In a case-sensitive language, "apple" and "Apple" are considered two different strings.

- Case Insensitive Programming Language

  In a case-insensitive language, the distinction between uppercase and lowercase letters is ignored when letters are compared. This means that 'A' is considered the same as 'a,' and two strings that differ only in their letter case are treated as equal. For example:

  In a case-insensitive language, "apple" and "Apple" are considered the same string.

- Case Sensitive-Insensitive Programming Language

  Some programming languages allow you to choose whether a comparison or search should be case-sensitive or case-insensitive. In these languages, you can specify whether you want the comparison to consider letter case or ignore it.

  Regarding Java, it is a case-sensitive programming language. Java considers uppercase and lowercase letters as distinct

3. Question 03

   Identity conversion is the process of converting a value of a reference type to the same type without any modification.

   Example:

   ```
   int num1 = 42;
    int num2 = num1;
   ```

4. Question 04

   Primitive Widening Conversion refers to the automatic and implicit conversion of a value from one primitive data type to another larger data type. This conversion occurs when the target data type can safely represent all possible values of the source data type without any loss of data.

   Example :

   ```
   byte myByte = 10;
   int myInt = myByte;
   ```

   The following diagram shows the ways of widening primitive conversions can happen between different data types.

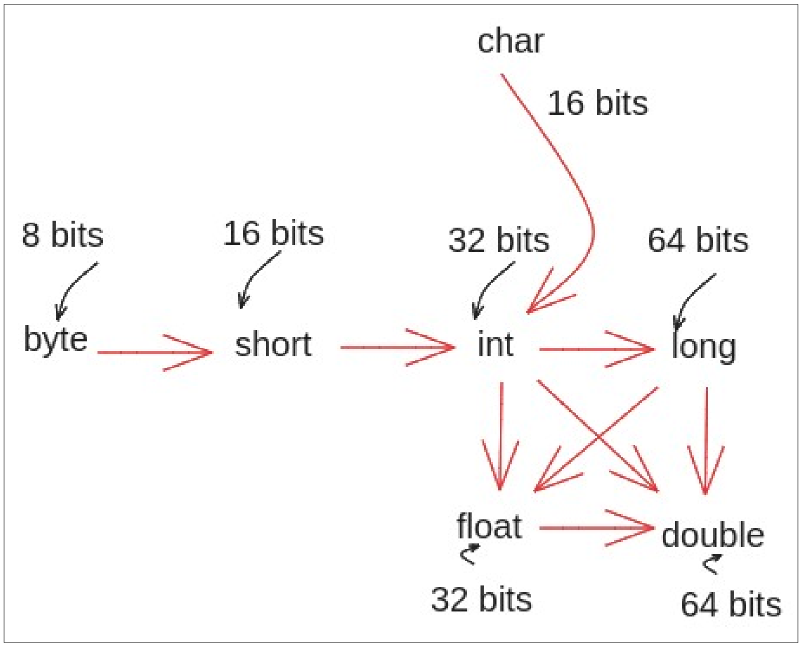

Figure 1: Diagram of primitive widening Conversion

5. Question 05

- Compile-time Constants

Compile-time constants are values that are known and determined by the compiler at the time of code compilation.

Example:

int myInt = 10;

- Runtime Constants

Run-time constants are values that are determined and assigned during the program's execution, specifically at runtime. Runtime constants are cannot be identified as constants at the compilation phase in a program.

Example:

final int MY-INT = (int)Math.random;

6. Question 06

   Implicit narrowing conversions are performed automatically by the compiler when assigning a larger data type to a smaller data type. However, explicit narrowing conversions (casting) are used when you want to explicitly allow data loss and inform the compiler about your intention. Implicit narrowing conversions can cause data loss if the value is not within the valid range of the target data type.

- Conditions must be met for an implicit narrowing primitive conversion to occur
  - The value being converted must fall within the valid range of the target data type.
  - The source must be a compile-time constant.

7. Question 07

   float data type in Java uses a special representation called the IEEE 754 floating-point format to store floating-point numbers. This format allows the float type to represent a wider range of values (positive and negative) at the cost of reduced precision compared to the long data type. In contrast, the long data type represents integers with a fixed 64-bit width.

8. Question 08

- int data type is the largest primitive data type which can be processed in a 32-bit system within one register. It takes 2 registers to process long data type (size – 64 bit) in 32-bit system. Because of that, int data type is set as the default data type to store integer literals in Java.
- The accuracy is considered as the most important factor in floating point literals. Double data type can have far more accuracy compared to float data type in java. Because of that, double data type is considered as the default data type for storing floating point literals in Java.

9. Question 09

   Implicit narrowing primitive conversion is limited to byte, char, int, and short because these data types have a defined size relationship and are commonly used interchangeably without significant data loss in many scenarios. They are small enough to safely fit into each other's ranges. Other types like long have larger sizes, making implicit narrowing impossible. Similarly, floating-point to integral conversion is not allowed implicitly due to their significant differences in representation.

10. Question 10

Widening and narrowing primitive conversion combines both widening and narrowing primitive conversions. This is used to convert byte to char. First, the byte is converted to an int via widening primitive conversion and then the resulting int is converted to a char by narrowing primitive conversion.