Programming Fundamentals - Assignment 2

Q1.

Statically Typed Language

languages which are consider about type at compile time are named as Statically Typed Languages.

Dynamically Typed Languages

languages which are consider about type at Run/Executing time are named as Dynamically Typed Languages.

Strongly Typed Language

These languages are Strongly consider about the type when assigning values

Loosely Typed Language

These languages are not Strongly consider about the type when assigning values to variables.

=> JAVA is Strongly typed Statically Typed and Dynamically Typed language.

Q2)

Case Sensitive

If language consider about the lover case and uppercase of the used letters in Identifiers, Key words in the source code are consider as case sensitive language.

Public,public => in Case Sensitive language these are two cases

Case Insensitive

If language doesn't consider about lover cases and uppercase in Identifiers when writing the source code that kind of language consider as case Insensitive language.

Cane use identifiers both as mySecond and mysecond.Case Sensitive -Insensitive

Language behave both ways of case sensitive and case Insensitive are consider as Case Sensitive - Insensitive Language.

=> JAVA consider as case sensitive language. JAVA consider “mySecondTry” and “mysecondtry” as two identifiers.

Q3)

Identity Conversion

this happen between same types. Free assign identifier can again to a same type identifier. When this happen its call Identity Conversion occurs.

Int x = 10;

int y =x;

jls 5.1.1. => A conversion from a type to that same type is permitted for any type.

Q4)

Primitive widening Conversion

1. occur between primitive data types

2. capacity of the type increase as byte<shote<char<int<long<float<double

3. concept says smaller capacity data type can convert to larger type

4. when value assign to a larger type, smaller type value convert to larger type

Q5)

Compile-time constant =>Value of these constants are determine by compiler in compile time

- final int x=10;

run-time constant => Compiler cant determine these values. Value of these constants are determine and assign in runtime.

- Final int x = abc(); => compiler compile this statement without assigning value to x, but in run time by this method value assign to x

Q6)

Implicit (Automatic) Narrowing Primitive Conversions1. Occurs If the larger type assigning value is with in the range of smaller data type.

2. occurs if the value is constant in compile time.

3. This happen only between byte,short, int,char

Explicit Narrowing Conversions (Casting)

1. this occur when going to assign larger type value to a smaller type.

2. if larger type value is not within the smaller type capacity range and still value assign to smaller type its call primitive data conversion (Casting) => this is defer

3. when this happen data loss can be occur.

Q7)

1. In long data type data saving method is deferent from float data saving method
2. Float type is design to store floating point values. If floating points are not considering last floating points it does not change the value in considerable amount. So in float type if its going save large amount of floating points this type is not consider last floating points. Because of this the value is not 100% accurate.
3. when long data type assign to float data type 64 bit structure change and the number is change as a floating point number. After change to a floating pint number if there are more than 23 floating points remove the last floating points and store the value in floating point structure. This can occur data lost and can be not 100% accurate.
4. This saving is happen as Exponent saving and Fraction saving.

Q8)

Long data type need 64 bit large space to store values. If long take as default it take unnecessary data space. If take byte or short there data capacity is not enough. So when When come to double it is mostly use to store large values. And accuracy level of double is higher than float. And can store more large values than float.

Q9)

Data range of byte, short, char and int is all most equal. there is no large gapes between capacity. But long is far more deferent. If long data type is able with narrow primitive conversion Data loss can be occur.

10)

Widening and narrowing conversion happen between bye and char when byte convert to char first widening occur byte to int and then narrowing between int to char

short to char is special case. char is unsigned 16 bit data type and short also 16 bit data type and when convert short to int it

convert to char is special way nor widening or narrowing.