8 Compare different types of variables in Java and discuss the scope and lifetime of variables?

Type		Scope	Lifetime
Local Variables	The variables declared inside a method or a block are known as local variables. A The local variable is created ~~to~~ when execution control enters the method and after the method completion	Within block (or) method the variables are visible and accessable	The variable life time is ended if method body execution is done
Instance Varibles	The variables declared inside a class and outside any method, constructor or block ~~are~~. These variables are visible to all the methods of the class. These variables ~~are created~~ create a seperate copy for every object of that class	within class the variables are visible and accessable	The variable life time is ended if class body execution is done

Static variable	These are declared using the static keyword. Satatic variable are initialized only once, at the start of program execution. The static variable only has one copy per class irrespective of how many objects we create	They can be accessed from anywhere within the class, including from static methods	They are destroyed when class is unloaded
Final variable	A final variable is a variable that declared using final keyword. The final variable is initialized only once, and does not allow any method to change it's value again. The variable created using final keyword acts as constant. all variables can be final variable	Same as any other variables	Same as any another variable

29 Java is a platform independent language. Justify your statement.

Ans Java is a platform-independent language because it uses a virtual machine (JVM). The JVM is a software program that translates Java bytecode into machine code that is specific to the underlying hardware platform. This means that Java programs can be compiled once and run on any platform that has a JVM installed.

Java bytecode is a plate form independent intermediate representation of the Java program. It is generated by the Java compilier and contains all of the information that the JVM needs to execute the program. This includes the instructions for the programe, as well as the data that the program needs.

The JVM is responsible for translating Java bytecode into machine code that is spectific to the hardware platform. It's done by Just in time (JIT) compilier.

Thus the use of a virtual machine make a Java a platform - independent language

3Q How to create and use user-defined packages?

Ans User defined Package:

Creating a ~~use~~ package: The user of the Java language can also create their own packages. ~~They are called~~ user defined packages can be also be imported into other classes and used exactly in the same way as built in packages

Syntax:

1. Package packagename
2. Package packagename.subpackage name;

Simple example: of

```
Package mypack;
public class simple {
    public static void main (String args[]){
        System.out.println("Welcome to package");
    }}
```

To compile a Java ~~program~~ package if you are not using any IDE, you follow the Syntax

Javac -d directory javafilename

Eg
javac -d . Simple.java

How to run java package program.
You need to use fully qualified name eg.
mypack.simple

To Run : java mypack.Simple
To Compile: javac -d . Simple.java

4Q Discuss the access control use in java. Explain with an example.

Ans Access control in Java is a mechanism that allows developers to restrict access to certain parts of their code. This can be done to protect sensitive data, prevent accidental errors and improve the overall maintainability of the code.
There are 4 access control modifier in Java:
* Public : The public modifier allows the code to be accessed from anywhere in the program
* Private : it allows the code to be accessed only from within the class in which it is declared
* protected : The protected modifier allows the code to be accessed from within the class in

which it is declared, as well as from subclasses of that class

* default : allows the code to be accessed from within the class in which it is declared, as well as from other classes in same package

Eg:

```
public class MyClass {
    private int count = 0;

    Public void inC C ){
        count ++;
    }
    public int gC C){
        return count;
    }
}
```

5Q Explain the usage of throw and throws with a suitable example for each

Ans 1) Throw: The keyword 'throw' is used to explicitly throw an exception. This can be done to indicate that an error has occured and that program cannot continue executing

Eg:

```
int divide (int a, int b) {
    if (b == 0) {
        throw new ArithmaticException("Division by 0");
    }
    return a/b;
}
```

2) Throws : The 'throws' keyword is used to declare that a method may throw one or more exceptions. This allows the caller of the method to be aware of the potential exceptions and to handle them accordingly

Eg:

```
public calss MC {
    public int divide (int a, int b) throws ArithmeticException {
        if (b==0) {
            throw new ArithmeticException("Division by zero");
        }
        return a/b;
    }
}
```