* Primitive data type
  + Numeric data type(signed data type)
    - Byte (1 byte = 8bits) (-128 +127)
    - Short(2 bytes =16bits)(2^15 to 2^15 -1)
    - Int(4 bytes 32bits)(-2^31 to 2^31 -1)
    - Long
    - Float
    - Double
* Byte – is the best choice for if we want to handle data streams either from or the network.(file supported form and network supported is byte)
* Short – this is the most rairly used data type in java
  + Character data type
    - Char
    - Boolean
* Serialization:
  + The process of writing state of object to a file it’s called serialization.
  + It is the process of converting an object from java supported form into either file supported form or network supported from.
  + By Using FileOutputStream and ObjectOutPutStream we can implement serialization
  + It is the process of converting an object from either file supported or network supported form into java supported form
* Transient Keyword
* Transient modifier applicable only for variables but not for the methods and classes
* If you don’t want to save particular variable to meet security constraints then we should declare that variable as transient
* While performing serialization jvm ignores original value of transient variable and save default to the file
* Method readObject() ,writeObject()
* In which order we serialize the object we will have desterilize in that order otherwise CCExceptio
* If we don’t know order of object in serialization(
* Transient vs. static
* Static variable is not a part of state and hence it won’t participate in serialization due to this declaring static variable as transient there is no use
* Transient vs. final
* Final variable will be participate in serialization directly by the value
* Hence declaring a final variable as transient there is no impact
* **Encapsulation in Java** is a mechanism of wrapping the variable and methods into single unit.
* Polymorphism means many forms.A single object can refer to the super-class or sub-class depending on the reference type which is called polymorphism
* An instance of a class is called an object. The object has state and behavior.
* An interface is an abstract type that is used to specify a behavior that classes must implement.

Class: All Java codes are defined in a Class. It has variables and method

Object: An instance of a class is called an object. The object has state and behavior.

Types of Design Patterns: 1. Creational 2. Structural 3. Behavioural A. Creational Design Pattern:

1. Singleton

2. Factory

3. Abstract factory

4. Builder

5. Prototype

B. Structural Design Patterns

6. Adapter

7. Composite

8. Proxy

9. Fly weight

10. Facade

11. Bridge

12. Decorator

C. Behavioural

13. Template Method

14. Mediator

15. Chain of Responsibility

16. Observer

17. Strategy

18. Command

19. State

20. Visiter

21. Iterator

22. Interpretor

23. Memento

== and equals method Difference

* == is used to address or reference comparison.
* Equals method also is used for comparison if you don’t override default behavior is reference comparison method is declared in object class.
* String, StringBuffer, StringBuilder

String:

* String is immutable because of you cannot change and modification in content it will return always new object
* String class equals method is for content comparison equals method is override
* String is most use that’s why java provide SCP(string constant pool)
* If content is fixe we should go for string if content keep on change then on every change new object will be created then we should not go with string

Immutable = not changeable or not allowed modification.

StringBuffer:

* StringBuffer is allowed to change and modification it is mutable.
* StringBuffer class equals method is for reference comparison equals method is not override
* StringBuffer is less use or no use that’s why java is not allowed to stringbuilder.
* If content is keep on change we should go for stringbuffer