ASSIGNMENT 12

GENERICS

1. What are generics in java?
Ans: Generics is a feature in java that allows for the creation of classes, interfaces and methods that can operate on a variety of data types.

2. What are the benefits of using generics in java?
Ans: it provides compile time type safety by enabling the specification of the data type of objects that a class or method can work with.

- Code reusability
- Improve readability

3. What is generics class in java?
Ans: Generic class in java is a class that can work with different types of data. It is implemented using a type parameter which are specified inside angle brackets(<>).

4. What is a Type Parameter in Java Generics?
Ans: A Type Parameter in Java Generics is a placeholder for the type of data that is used by a generic class or method. It is defined using a single uppercase letter enclosed in angle brackets, such as<T>, <N>.

5. What is a generic method in java?
Ans: Generic method in java is a method that can work with different types of data.

6. What is the difference between ArrayList and ArrayList<T>?
Ans: ArrayList is a non-generic class and ArrayList<T> is a generic class.
ArrayList can store any type of element and ArrayList<T> can store only specified elements.

IO File Handling

1. What is Input and Output Stream in Java?
Ans: A stream can be defined as a sequence of data. The InputStream is used to read data from a file and the OutputStream is used for writing data to a file.

2. What are the methods of OutputStream?
Ans: write() - writes the specified byte to the output stream
flush() - forces to write all data present in the output stream to the destination
close() - closes the output stream

3. What is serialization in Java?
Ans: The process of saving or writing state of an object to a file is called serialization. It is the process of converting an object from java supported form to file supported form.

4. What is the Serializable interface in Java?
Ans: The Serializable interface in Java is a marker interface. It is used to mark classes that can be serialized, meaning their object instances can be converted into a stream of bytes.

5. What is deserialization in Java?
Ans: The process of reading state of an object to a file is called deserialization. It is the process of converting an object from file supported form to java supported form.

6. How is serialization achieved in Java?
Ans: Serialization is achieved in Java by implementing the Serializable interface. When an object is serialized, its state is converted into a stream of bytes, which can then be transferred over a network or stored in file.

7. How is deserialization achieved in Java?
Ans: Deserialization is achieved in Java by reading a stream of bytes and using them to recreate the original object instance. This is done by calling the readObject() method of an ObjectInputStream instance.

8. How can you avoid certain member variables of class from getting Serialized?
Ans: by using transient keyword we can avoid certain member variables of a class from getting serialized.

9. What classes are available in the Java IO File Classes API?

Ans: File

FileInputStream

FileReader

FileOutputStream

FileWriter

10. What is Difference between Externalizable and Serialization interface ?

Ans:

Serializable	Externalizable
A serializable interface is used to implement serialization.	An externalizable interface used to implement Externalization
Serializable is a marker interface i.e. it does not contain any method.	The externalizable interface is not a marker interface and thus it defines two methods writeExternal() and readExternal().
Serializable interface passes the responsibility of serialization to JVM and the programmer has no control over serialization, and it is a default algorithm.	The externalizable interface provides all serialization responsibilities to a programmer and hence JVM has no control over serialization.
Serialization using a serializable interface has bad performance.	Serialization using an externalizable interface has better performance.
Default serialization does not require any no-arg constructor.	A public no-arg constructor is required while using an Externalizable interface.
It is hard to analyze and modify class structure because any change in structure may break serialization.	It is relatively easy to analyze and modify class structure because of complete control over serialization logic.
Using a serializable interface we save the total object to a file, and it is not possible to save part of the object.	Base on our requirements we can save either the total object or part of the object.
Transient keywords play an important role here.	Transient keywords won’t play any role.