File Handling

Creating Directory, File, Writing into file and Reading from file

1. Steps to create Directory:

a. Import file class
b. Create an object of file
c. Check if it exist
d. Call mkdirs() function

```
    public static void main(String[] args)
    {
        File f=new File("D:\\Sample\\Prog");
        boolean b1=f.exists();
        if(b1!=true)
        {
            boolean b2=f.mkdirs();
            if(b2==true)
            {
                System.out.println("Directory is created");
            }
            else
            {
                System.out.println("Directory already exists");
            }
        }
    }
```

2. Creation of file

a. import file class
b. Create an object of file class
c. Verify whether the file already exists using exists function.
d. Create file using CreateNewfile function

```
public static void main(String[] args)
        {
            File f=new File("D:\\Ash\\Prog\\sample.txt");
            boolean b=f.exists();
            if(b!=true)
            {
                try
                {
                    boolean b1=f.createNewFile();
                    if(b1==true)
                    {
                        System.out.println("File is created");
                    }
                }
                catch (IOException e)
                    {
                    e.printStackTrace();
                    }
            }
        }
```

3. Write data into file:

a. import java.io.FileWriter
b. Create filewriter object
c. Call write function
d. call flush function using filewrite object

```
public static void main(String[] args)
        {
                try {
                        FileWriter fw= new FileWriter("D:\\Ash\\Prog\\sample.txt");
                        fw.write("Hello");
                        fw.flush();
                        }
                catch (IOException e)
                {
                        e.printStackTrace();
                }
        }
```

4. Reading from file

a. Import java.io.*;
b. Create FileReader object
c. Surround with try catch block
d. Call read function, which stores an integer value
e. Check if read value is not equal to -1 using while function(Check for end of line/ blank).
f. Print the character and again verify for -1 i.e blank/ space

```
public static void main(String[] args)
        {

        try {
                FileReader f=new FileReader("D:\\Ash\\Prog\\sample.txt"); // Add IO
exception
                int x=f.read();

                while(x!=1)
                {
                        System.out.print((char)x);
                        x=f.read();
                }
                }
                catch (Exception e) // to catch multiple exception make use of Exception
class
                {
                e.printStackTrace();
                }
}
```

```
public static void main(String[] args)
        {
                try
                {
                FileReader f=new FileReader("D:\\Ash\\Prog\\sample.txt"); // Add IO
exception
                int x=f.read();
                while(x!=-1)
                {
                        System.out.print((char)x);
                        x=f.read();
                }
                }
                catch (Exception e) // to catch multiple exception make use of Exception
class
                {
                e.printStackTrace();
                }
}
```

Serialization and De-serialization

Serialization is the process of storing java objects into the file is called as serialization. Serializable can be achived using serialization marker interface.

De-serializatioin

The process of reading serializable object from the file is known as deserializatiomn.

Serialization Steps:

1. Inherit a serializable marker interface (Interface which doesn’t contain a method/empty)
2. Create object of the inherited class.
3. Create fileSTream object (build stream btw java and file)
4. Create object of objectstream class.(Send data btw object file and file)
5. Write using objectstream class into the object of serialized method.
6. Flush the object.

```
public static void main(String[] args)
        {
                Sample s=new Sample();
                try {
                        FileOutputStream f=new
FileOutputStream("D:\\Ash\\sample.ser"); // create a stream btw java and file
                        ObjectOutputStream o=new ObjectOutputStream(f); // sends
object btw java heap and file
                        o.writeObject(s);
                        o.flush();// store object permanently
        }
```

De-Serialization

1. Create a class, implement a serializable marker interface.
2. Create fileoutputstream object.
3. Create object of objectoutputstream
4. To read use readobject(), with help of objectoutputstream
5. Downcast the upcasted object
6. Try to print object value using downcast object.

```
public static void main(String[] args)
    {
        try {
            FileInputStream f=new FileInputStream("D:\\Ash\\sample.ser");
            ObjectInputStream o=new ObjectInputStream(f);
            //upcasting
            Object ob=o.readObject();
            System.out.println(ob);
            //downcasting
        Sample s=(Sample)ob;
            System.out.println(s);
            System.out.println(s.x);

        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
```