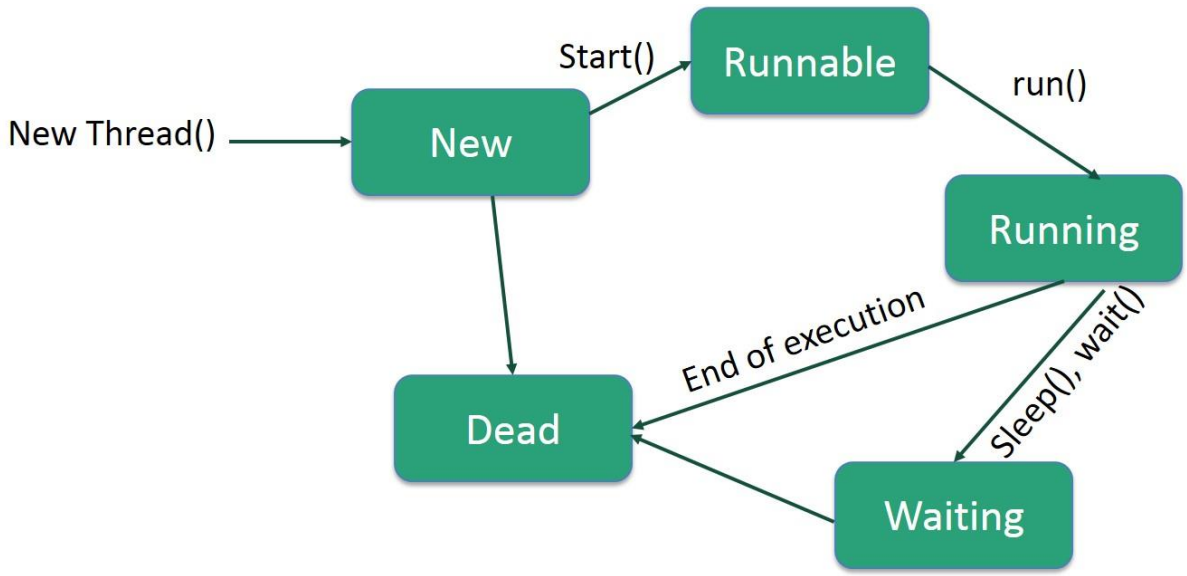
Wrapper classes

```
1. //Java Program to convert all primitives into its corresponding
2. //wrapper objects and vice-versa
3. public class WrapperExample3{
4. public static void main(String args[]){
5. byte b=10;
6. short s=20;
7. int i=30;
8. long l=40;
9. float f=50.0F;
10. double d=60.0D;
11. char c='a';
12. boolean b2=true;
13.
14. //Autoboxing: Converting primitives into objects
15. Byte byteobj=b;
16. Short shortobj=s;
17. Integer intobj=i;
18. Long longobj=l;
19. Float floatobj=f;
20. Double doubleobj=d;
21. Character charobj=c;
22. Boolean boolobj=b2;
23.
24. //Printing objects
25. System.out.println("---Printing object values---");
26. System.out.println("Byte object: "+byteobj);
27. System.out.println("Short object: "+shortobj);
28. System.out.println("Integer object: "+intobj);
29. System.out.println("Long object: "+longobj);
30. System.out.println("Float object: "+floatobj);
31. System.out.println("Double object: "+doubleobj);
32. System.out.println("Character object: "+charobj);
33. System.out.println("Boolean object: "+boolobj);
34.
35.
```

```
36. //Unboxing: Converting Objects to Primitives
37. byte bytevalue=byteobj;
38. short shortvalue=shortobj;
39. int intvalue=intobj;
40. long longvalue=longobj;
41. float floatvalue=floatobj;
42. double doublevalue=doubleobj;
43. char charvalue=charobj;
44. boolean boolvalue=boolobj;
45.
46. //Printing primitives
47. System.out.println("---Printing primitive values---");
48. System.out.println("byte value: "+bytevalue);
49. System.out.println("short value: "+shortvalue);
50. System.out.println("int value: "+intvalue);
51. System.out.println("long value: "+longvalue);
52. System.out.println("float value: "+floatvalue);
53. System.out.println("double value: "+doublevalue);
54. System.out.println("char value: "+charvalue);
55. System.out.println("boolean value: "+boolvalue);
56. }}
```

Thread Life Cycle



Multithreading Programs

```
public class Table {
        synchronized void printTable(int n){//synchronized method
                for(int i=1;i<=5;i++){
                  System.out.println(n*i);
                  try{
                   Thread.sleep(400);
                  }catch(Exception e){System.out.println(e);}
                }
 }

}

public class MyThread1 extends Thread {
        Table t;
        MyThread1(Table t){
        this.t=t;
        }
        public void run(){
        t.printTable(5);
        }
}
```

```
public class MyThread2 extends Thread {
	Table t;

	MyThread2(Table t){
	this.t=t;
	}

	public void run(){
	t.printTable(100);
	}

}


public class Main {
	public static void main(String args[]){
		Table obj = new Table();//only one object
		MyThread1 t1=new MyThread1(obj);
		MyThread2 t2=new MyThread2(obj);
		t1.start();
		t2.start();
		}
}
```

File Handling

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

//Class to demonstrate File Creation , File writing & File Reading
public class FileHandling {

        public static void main(String args[]) {
      try {
            // Creating an object of a file
            File f0 = new File("F://FileExample.txt");
            if (f0.createNewFile()) {
                  System.out.println("File " + f0.getName() + " is created
successfully.");
            } else {
                  System.out.println("File  already exists in the directory.");
            }
         } catch (IOException exception) {
                System.out.println("An unexpected error is occurred.");
                exception.printStackTrace();
        }

      //Writing to the above file
      FileWriter fwrite;
                try {
                      fwrite = new FileWriter("F://FileExample.txt");
                       // writing the content into the FileExample.txt file
               fwrite.write("A named location used to store related information is
referred to as a File.");

               // Closing the stream
               fwrite.close();
               System.out.println("Content is successfully wrote to the" );
                } catch (IOException e) {
                      // TODO Auto-generated catch block
                      e.printStackTrace();
                }

                //Reading from the above file
```

```
            try {
        // Create f1 object of the file to read data
        File f1 = new File("F://FileExample.txt");
        Scanner dataReader = new Scanner(f1);
        while (dataReader.hasNextLine()) {
            String fileData = dataReader.nextLine();
            System.out.println(fileData);
        }
        dataReader.close();
    } catch (FileNotFoundException exception) {
        System.out.println("Unexcpected error occurred!");
        exception.printStackTrace();
    }

 }
}
```