1. thread is a small chunk of a big process

2. we can create thread in java 2 ways

a. extends Thread class

b. implement Runnable interface

3. the logic that needs to be executed is written inside a overridden method i.e run() method

//multi level inheritance

class A extends Thread {

public void run() {

for (int i = 0; i < 10; i++) {

System.out.println("----a---" + i);

}

}

}

class B implements Runnable{

public void run() {

for (int i = 0; i < 10; i++) {

System.err.println("----b---" + i);

}

}

}

4. we should not directly call the run method

A a = new A();

B b = new B();

a.run(); //wrong

b.run(); //wrong

5. we should call start method instead of directly calling run method

6. we cannot directly call start method inside runnable thread - because we dont have start method inside runnable interface

A a = new A();

a.start();

B b = new B();

a.start(); //wrong

b.start(); //start method is not available

7. for runnable threads we need to create instance of thread class and call start method on it.

B b = new B();

Thread th = new Thread(b);

th.start();

Thread life cycle ->

new

runnable

running -> blocked

terminated

windows -> priority based processing thread;

unix/linux/ mac -> round and robin based