Lesson 6 : packages and multi threading

Multi threading :

Program : set of instruction to perform a specific task.

Processor :Processor is responsible to execute the code.

Process : time taken to execute the code is known as process.

Thread : Thread is a small execution of process.

Thread also known as light weighted process. It takes less memory or less resources to do same task.

By default java is thread based programming language.

By default C or C++ is process based programming language

Once we give java programs to execute internally java use default thread to execute that code.

Inside a main method default thread always execute.

If we want to find default thread details we need to take the help of Thread class.

Thread is a pre defined class part of lang package.

```
Thread t = Thread.currentThread();
```

currentThread is a pre defined method part of thread class and it is a static method and method return type is same class reference.

System.out.println(t); Thread[main,5,main];

main -→ name of the thread

5-→ priority of the thread

Main -→ group of the thread.

t.setName(“MyThread”);

t.setPriority(2); min priority 1, max 10 default norm 5

multi tasking : more than one task at time.

We can achieve multi tasking using

1. Process based
2. Thread based

2 time to process

Copy set of files from one drive to another drive

1st Task	10 min	C	To	D
2nd task	10 min	D	To	E
3rd task	10 min	E	To	F

Malti tasking using thread based is faster than multi tasking using process based.

Java support auto GC : Garbage collector.

In java we can create custom or user defined thread using two ways

1. Extends Thread class
    a. First we need to create user defined class and that class must be extends Thread class.
    b. Now we need to create thread class reference ie object create for thread.
    c. Using thread class reference call start() method. Start() is a pre defined method part of thread class which help to make thread is ready to run ie Runnable state.
    d. start() method internally call run() method. run() method is a part of thread class. pre defined run() method doesn't contains any logic.
    e. If we want to perform some custom task then we need to override run() method part of thread class.
2. Implements Runnable interface.
    a. First we need to create user defined class and that class must be implements Runnable interface.
    b. Runnable is a pre defined interface which contains one method is run(). When our class implements runnable interface we need to override run method mandatory.
    c. Run method purely belong to runnable interface. Thread class internally implements Runnable interface and provided empty body for run method.
    d. Now we need to create Runnable interface reference.
    e. Using that reference we can't call start() method. because start method is part of thread class.

Life cycle of thread

Create ----------→ ready to run ----------→ running ---------→ destroy

Runnable

obj1 obj.start() run i=10

t1 t1.start() run i=10

isAlive()

sleep()

wait()

notify()

notifyAll()

sleep() method is use to pause the thread flow. It can be pre defined or user defined.

isAlive() : this method is use to check the status of thread.

10 thread in one task. Then we need to create only one task class and create more than one thread.

Number of task equal number of classes.

Transfer the amount : task

Book the ticket : task

Check balance : task

Place the order : task

Synchronization : it is use to block or lock the thread.

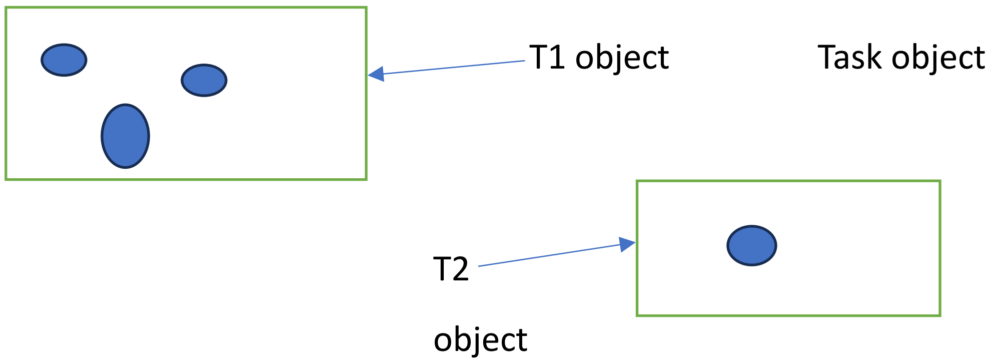
Using synchronization we allow only one thread to use all resources at time.

To achieve synchronization we need to use synchronized keyword.

This keyword we can use with method as well as inside a method we can

Use more than one synchronized.

Inner thread communication



Pts

1. More than one thread must be part of same memory.
2. Method must be synchronized.
3. Using wait() method we can make the thread to wait or suspend.
4. Using notify() method we can callback waited thread or resume the thread.
5.

wait()

notify()

notifyAll()