UNIT - 4

**MULTITHREADING AND GENERIC PROGRAMMING**

Question Bank

**PART – A**

**2 – Marks**

1. **Define Threads.**

A thread is a lightweight sub-process, the smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking. It shares the same memory area. Context switching between thread is faster.

1. **List the life cycle stages of thread.**

* Create
* Runnable
* Running
* Non Runable
* Terminated

1. **What are all the advantages of usages of threads?**

It doesn't block the user because threads are independent and you can perform multiple operations at the same time. We can perform many operations together, so it saves time. Threads are independent, so it doesn't affect other threads if an exception occurs in a single thread.

1. **Define multitasking system.**

In multitasking system Each program can have n number of task’s. Similar to multi programming but a part of a program [task] gets executed instead of whole program. Single CPU performs multiple task based context switching. Task is sub unit for whole program can be compared to single thread.

1. **Give a short note on Runnable and Running stages of thread.**

***Runnable:***

The thread is in runnable state after invocation of start() method, but the thread scheduler has not selected it to be the running thread. It simply announces to scheduler that thread is ready to run.

***Running:***

It is post runnable thread, here the runnable thread is selected to thread scheduler and started to run already. It remains in running state until some interruption event has occurred.

1. **Write a program that prints odd numbers only upto the limit using threads.**

class PrintNumbers extends Thread{

String threadName;

int limit;

public PrintNumbers(String name, int limit){

this.threadName = name;

this.limit = limit;

}

public void run() {

for(int i=1;i<=limit;i++) {

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

}

}

}

public class Main {

public static void main(String[] args) throws InterruptedException {

PrintNumbers t1 = new PrintNumbers("t1",5);

t1.start();

}

}

***Output:***

t1--1

t1--2

t1--3

t1--4

t1--5

1. **Give a short note on synchronization of thread.**

Multi-threaded programs may often come to a situation where multiple threads try to access the same resources and finally produce erroneous and unforeseen results. This is known as problem of synchronization we need take care and impose some regulation for accessing shared resource. This situation is referred to race condition.

1. **List the types of synchronizations.**

There are two types of synchronizations are available

* ***Mutual Exclusion***
  + Synchronized Methods.
  + Synchronized Block.
  + Static Synchronization.
* ***Cooperation***
  + Inter thread communication

1. **What is co-operative thread?**

If a process contains more than one thread and it is synchronized for process completion states then the threads are known as co-operative threads. In co-operative scenario the co-operating threads depends on each other for successful completion of one process.

1. **What is demon thread?**

Daemon thread in java is a service provider thread that provides services to the user thread. It provides services to user threads for background supporting tasks. It has no role in life than to serve user threads. Its life depends on user threads. It is a low priority thread.

1. **What is Java Generics?**

Java generics help us to create classes and methods that independent of type and also it ensures the compile time type safety. Java generics available from J2SE version 5.0.

1. **Give an example for generic methods.**

class GenericArrays{

<E> void printArray(E[] aArray) {

System.out.println("Object Types:"+aArray.getClass().getName());

for(E temp:aArray) {

System.out.println(temp);

}

}

}

public class Main {

public static void main(String[] args) {

GenericArrays a = new GenericArrays();

Integer[] i = {1,2,3,4};

a.printArray(i);

Float[] f = {1.3f,2.55f,3.56f,4.67f};

a.printArray(f);

String[] s = {"Apple", "Boy","Cat","Dog"};

a.printArray(s);

}

}

***Output:***

Object Types:[Ljava.lang.Integer;

1

2

3

4

Object Types:[Ljava.lang.Float;

1.3

2.55

3.56

4.67

Object Types:[Ljava.lang.String;

Apple

Boy

Cat

Dog

1. **What is bounded types in generics?**

When we handling generics then we can restrict the type parameter to only receive certain types. For example if we need to perform addition of array elements then it means the array contains only numbers. Performing summation of arrays doesn't make sense to string arrays. So we can restrict the type to numbers.

1. **List two restrictions of generics.**

* Cannot Instantiate Generic Types with Primitive Types
* Cannot Create Instances of Type Parameters
* Cannot Declare Static Fields Whose Types are Type Parameters

1. **Why static members of types not supported in generics**?

Static types are class variables. But generic types are decided based on type class that specified. When we define a variable as static it must be common to all type and it must accessible with Class name. Let us consider generic class has static variable but in has instance of multiple types for eg Integer, Float, String ... etc. Then it raises one problem that which type we need to keep as common for all. Technically it is not possible. So static members of types not supported in the generics.

**PART – B**

**13 -Marks**

1. Write a program to find the generated Fibonacci series number is prime number or not.
2. Explain about synchronized methods with example.
3. Explain about synchronized blocks with example.
4. Explain about co-operative process with example.
5. Explain about thread groups with an example.
6. Write a program that helps to find all sum of all elements in array using generics.
7. Explain in detail about generic class and methods.