OLQ12

Study Guide

OLQ12 will be multiple choice, true/false and fill in the blank questions over the following topics/vocabulary.

Superclass/Subclass

Polymorphism

Inheritance

Encapsulation

Abstraction

Exceptions

- Checked vs Unchecked
- RuntimeException
- Error
- try-catch-finally
- assert

Constructors

- order that constructors are called in

Access specifiers – public, private, protected

Primitive types and reference types

UML

abstract class vs concrete class

interfaces

concurrent vs parallel

thread vs process

thread states

`Executor`

`Runnable`

thread pool

`ExecutorService`

final and multithreading

`InterruptedException`

ordering of threads

Thread synchronization

Monitors

Type wrapper classes

Autoboxing and auto unboxing

OLQ12 will also require writing code to start multiple threads that use an object to accomplish a task. We did multiple examples in class. Here is one of them

```
public class ConcurrencyDemo
{
    public static void main(String[] args)
    {
        ArrayList <NumberThread> NT = new ArrayList<>();

        for (int i = 10; i < 100; i++)
        {
            NT.add(new NumberThread(i));
        }

        System.out.println("Starting Executor");

        ExecutorService executorService = Executors.newCachedThreadPool();

        for (int i = 0; i < 90; i++)
        {
            executorService.execute(NT.get(i));
        }

        executorService.shutdown();

    }
}
```

```
package concurrencydemo;

public class NumberThread implements Runnable
{
    private int number;
    static int counter = 1;

    public NumberThread(int number)
    {
        this.number = number;
    }

    public void run()
    {
        synchronized (System.out)
```

```
        {
            System.out.printf("%d-", number);

            if (counter++ % 10 == 0)
            {
                System.out.println();
            }
        }
    }
}
```