WELCOME & THANK YOU



Advance Java Crash Course

For TD Bank

MEET YOUR CRASH COURSE TEAM



TANGY F.
CEO



WILLIAM D. DEVELOPER

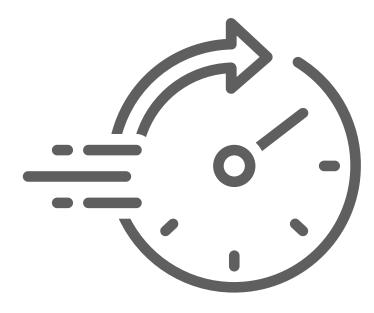


TONY J.

DEVELOPER







Rapid Review TRIVIA

Rapid Trivia

Which company, in the 70s created a research center in Palo Alto,CA that developed

- The local area network
- The graphical user interface
- Heavily contributed to object oriented programming concepts.



TODAY'S AGENDA

TRAINING
DAY 2

1 Yesterday's Coding Exercise to Go Rapid Review

2 Heap Memory Errors

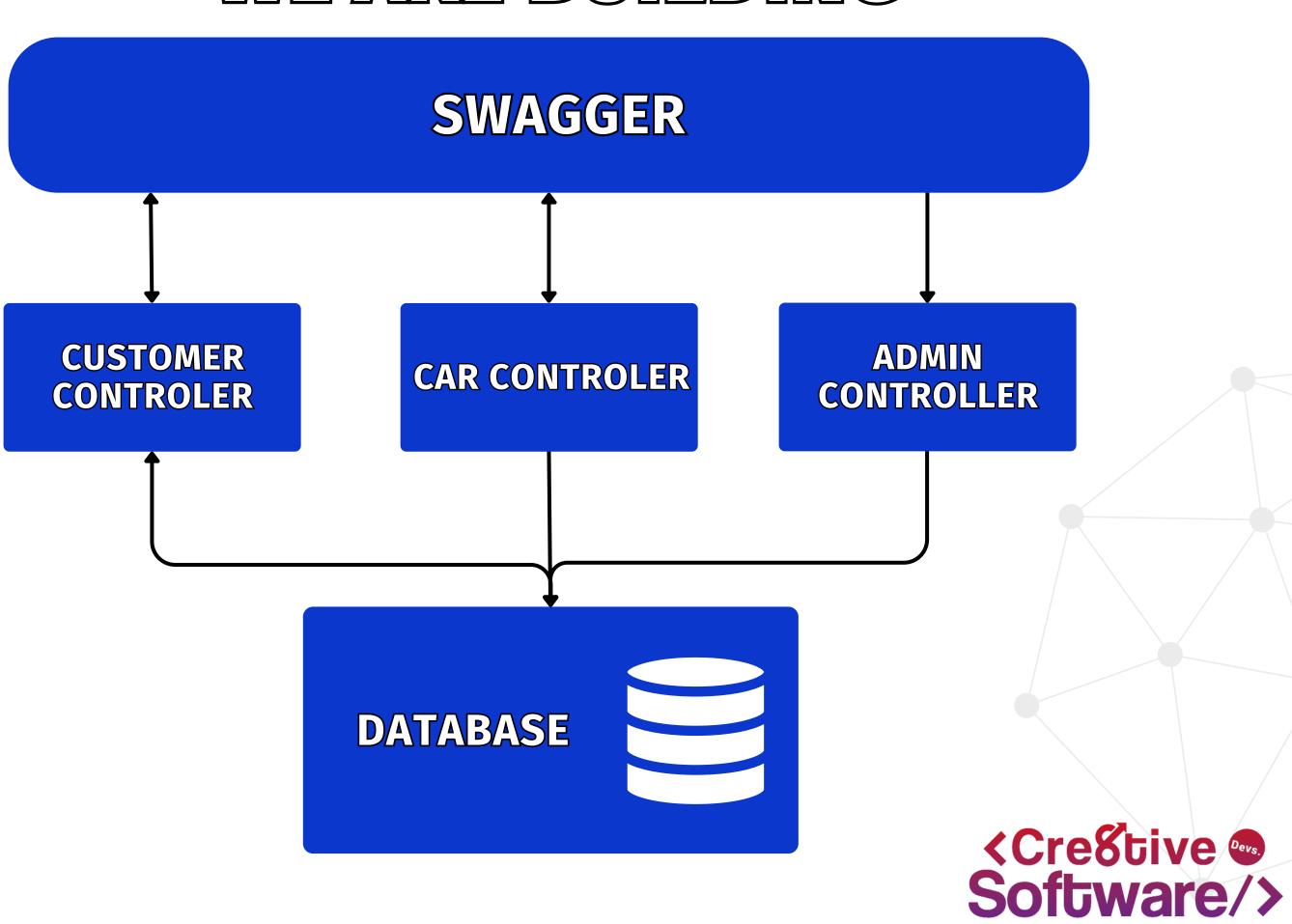
Transaction Management

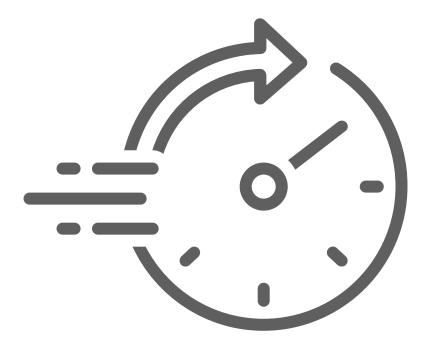
4 Multithreading



WE ARE BUILDING

BITE SIZE
CAR RENTAL APP



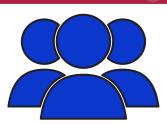


Rapid Review Lesson 1

Rapid Review Adding Year Validation to Add Car



YERSTERDAYS CODING EXERCISE TO GO



In our **admin-controler /admin/addcar,** when we add zero to "year", it is not returning "Invalid car details" as expected. Find the right location in the code and make the necessary correction.

Hint: Go to the **AdminController.java** (bellow), place a brakepoint in line 25 and step inside **isValidCarRequestDto** and study the execution to find out.

```
POST /admin/addcar

Parameters

No parameters

Request body required

{
    "manufacturer": "Mazda",
    "model": "Model 6",
    "year": 0,
    "color": "Red",
    "price": 223,
    "isBooked": false
}
```

```
@PostMapping(value = "addcar")
public ResponseEntity<? extends AbstractCarDto> addCar(@RequestBody CarRequestDto) throws IllegalAccessException {
    if (!CarRequestValidator.isValidCarRequestDto(carRequestDto)) {
        throw new InvalidArgumentException(message:"Invalid car details");
    }
    CarResponseDto createdCar = carService.addCar(carRequestDto);
    return new ResponseEntity<>(createdCar, HttpStatus.CREATED);
}
```

YESTERDAY CODE TO GO

com.carrental.autohire.dto.CarRequestDto

```
package com.carrental.autohire.dto;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.EqualsAndHashCode;
import lombok.NoArgsConstructor;
@Data
@AllArgsConstructor
@NoArgsConstructor
@EqualsAndHashCode(callSuper=false)
public class CarRequestDto extends AbstractCarDto {
    private Boolean isBooked = Boolean.FALSE;
    private String manufacturer;
   private int year; // Missing
```



Let's see it running.



Heap/Memory Errors



HEAP AND MEMORY ERRORS

Heap

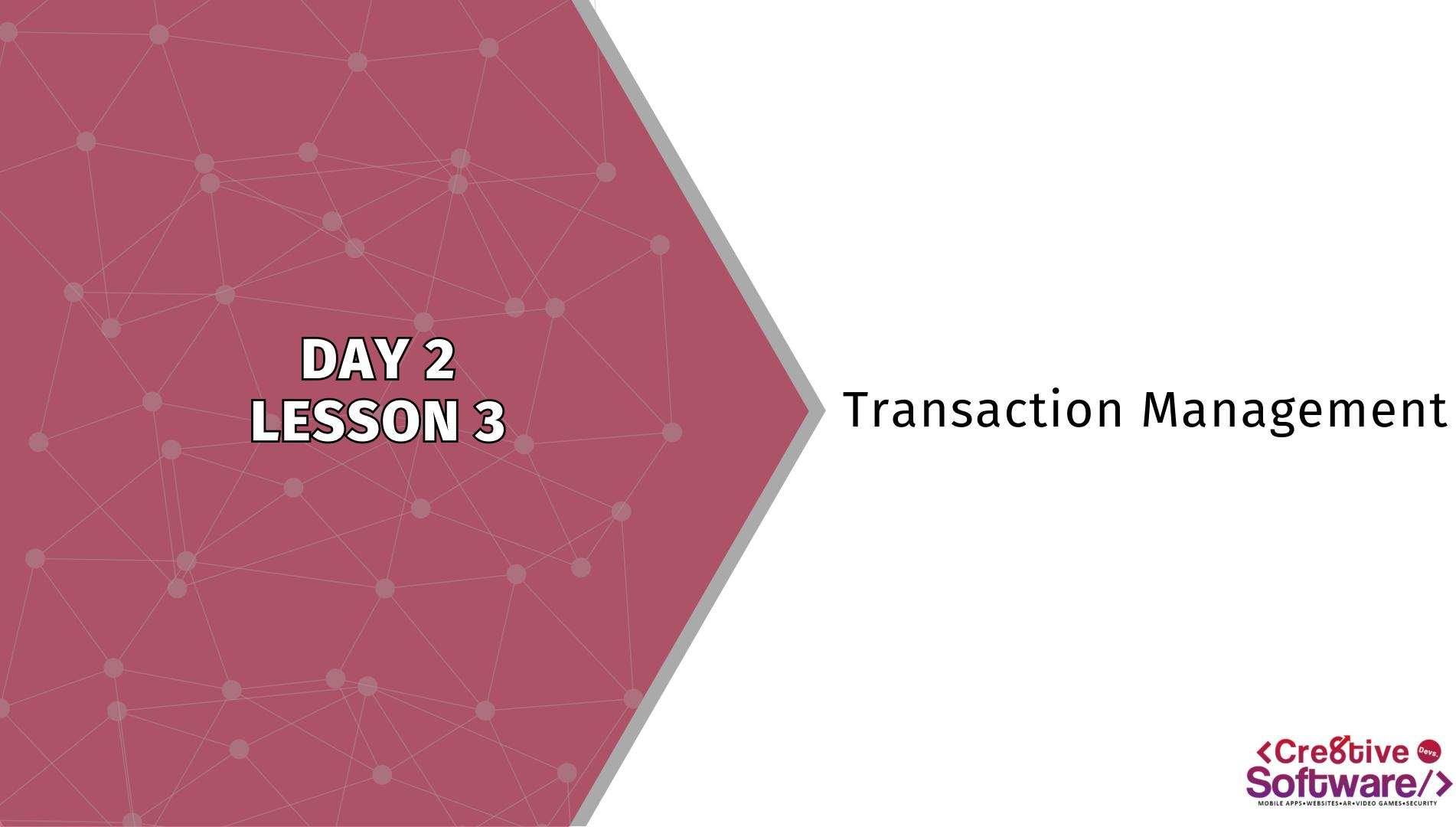
- Region responsible for dynamically allocating and deallocating Objects.
- Can grow and shrink as needed.
- Garbage collector periodically identifies and deallocates objects not in use.
- Can be adjusted through JVM configuration parameters.
- Correctly allocating heap area to our projects prevents OutOfMemoryError

Avoid Memory Errors

- Use JVM options like -Xmx and -Xms, to correctly allocate heap size.
- Use memory usage tools like JVisualVM, JConsole, or Java Mission Control.
- Close Resources Appropriately.
- Use StringBuilder rather than string concatenation.
- In batch processing, breakup large amount of data into smaller portion. You can use ParallelStreams for that (we'll see on day 5)
- Avoid creating too many threads, use thread pools.



Let's see the code.



TRANSACTION MANAGEMENT

Commitment Control

Set AutoCommit to false before any output operation.

```
connection.setAutoCommit(autoCommit:false); // Dissable autoCommit
```

Rollback if an output operation fails.

```
int rowsAffected = insertCustomerCarStatement.executeUpdate();
if (rowsAffected == 1) {
    log.info(format:"Added entry to car_customer table for Customer ID: {} and Car ID: {}", customerId, carId);
} else {
    log.error(msg:"Failed to add entry to car_customer table");
    connection.rollback(); // Roll back because of error
    return null;
}
```

Commit after all output operations have succeeded.

```
carStatement.setBoolean(parameterIndex:1, x:true);
carStatement.setLong(parameterIndex:2, carId);
carStatement.executeUpdate();

connection.commit(); // commit transaction
```



Let's see it running.



Multithreading



MULTITHREADING

```
class PrintThread extends Thread {
    private final int start;
    private final int end;

public PrintThread(int start, int end) {
        this.start = start;
        this.end = end;
    }

@Override
public void run() {
        for (int i = start; i <= end; i++) {
            | System.out.println("Thread " + Thread.currentThread().getId() + ": " + i);
        }
    }
}</pre>
```

```
PrintThread thread1 = new PrintThread(start:1, end:5);
PrintThread thread2 = new PrintThread(start:6, end:10);
thread1.start();
thread2.start();
```

```
Runnable emailTask = () -> sendEmail(email, subject, body, receiptFileName);
Thread emailThread = new Thread(emailTask);
emailThread.start();
```



Let's see it running.



CODING TRIVA QUESTION #1



What are ternary operators in Java?



CODING TRIVA ANSWER #1



Ternary operators provide abbreviated syntax for if then else conditions using the following format;

```
variable = (condition) ? whentrue : whenfalse
```

return (condition) ? whentrue : whenfalse



CODING TRIVA QUESTION #2



Convert the following if then else code to use ternary.

```
if (car != null) {
    return new ResponseEntity<>(car, HttpStatus.OK);
} else {
    return new ResponseEntity<>(HttpStatus.NOT_FOUND);
}
```



CODING TRIVA ANSWER #2



Without Ternary

```
if (car != null) {
    return new ResponseEntity<>(car, HttpStatus.OK);
} else {
    return new ResponseEntity<>(HttpStatus.NOT_FOUND);
}
```

With Ternary

```
return (car != null) ? new ResponseEntity<>(car, HttpStatus.OK) : new ResponseEntity<>(HttpStatus.NOT_FOUND);
```



CODING EXERCISE TO GO

In our car-controler /api/cars/unregister by carld the release of a car involves 2 tables with executeUpdate(). Find the location where this happens and add commitment control to ensure they are not left out of sync if the second executeUpdate() fails for any reason.

You will need to book a car first and then release it with the unregister API

car-controller	^
POST /api/cars/unregister/{carId}	^
Parameters	Try it out
Name Description	
carld * required integer(\$int64) (path) carld	





Crash Course

We will see you tomorrow

