

2017

31 Jan / 2022

January
DAY - 1 Sunday

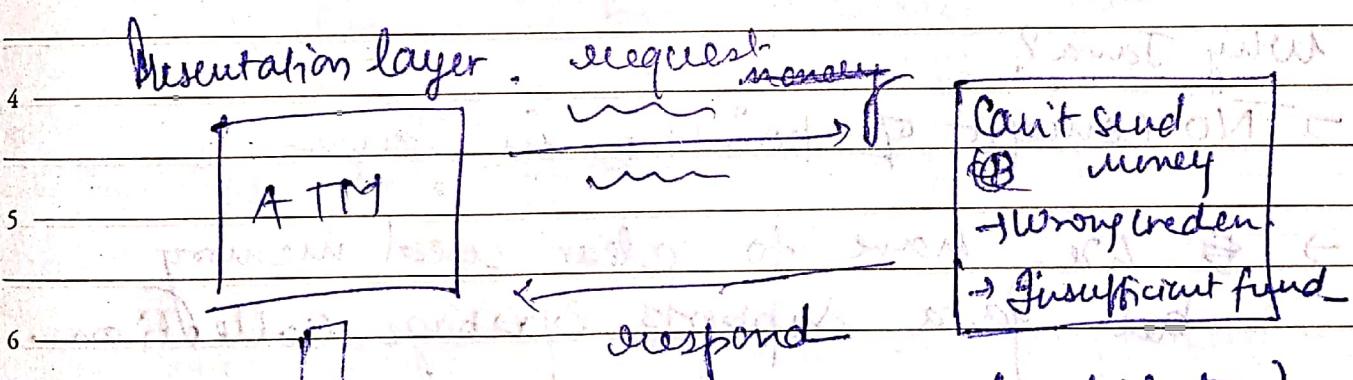
UI

Week-53 • Day : 001-364

- 8 Java Runtime Version: `java -version`
 9 Java Compiler Version: `javac -version`

- 10
 11 → Consumer Electronics i Embedded system based application. (Remote, car)
 12 → Enterprise level application.

- 1
 2 Presentation layer: Presenting UI by which user can interact.



Evening

(pti) industry

20	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	E
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

FEB

MAR

APR

MAY

JUN

Java Supports :

- stand alone application : only one person can access application at a time.
- console based " (Command line interface)
- Graphical " "
- Enterprise level " : Single application can be accessible by many people.

~~Properties~~

Programming language

A language in which we write out our business logic and is understandable by our system.

Why Java?

- No concept of pointers in Java
- We have to clear used memory but Java supports Garbage Collection.
- Can write streamlined & clear code.
- Speed the dev., &
- Code reusability (portability). Write once, used many times.

S	A	S	H	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

jdb → java debugger.

2017

January

Tuesday

03

Week-01 • Day : 003-362

- It is development environment:
provides us compiler, environment to write down, debug, testing, documentation. everything required for SDLC.
- It is an Application environment.
- It is an deployment envt: (Accessible).
sitting at our own system we can deploy enterprise level app. accessible by many users.
- Applets: presentation layer (UI).
- Application: Enterprise level application.

Evening

8 Goals of Java

- 9 → Multiple snap of codes simultaneously: Threading.
- 10 → loads classes dynamically at the time they are required.
- 11 → Better security, doesn't depends on OS, used its own runtime envt.

2 Features fulfilling goals:

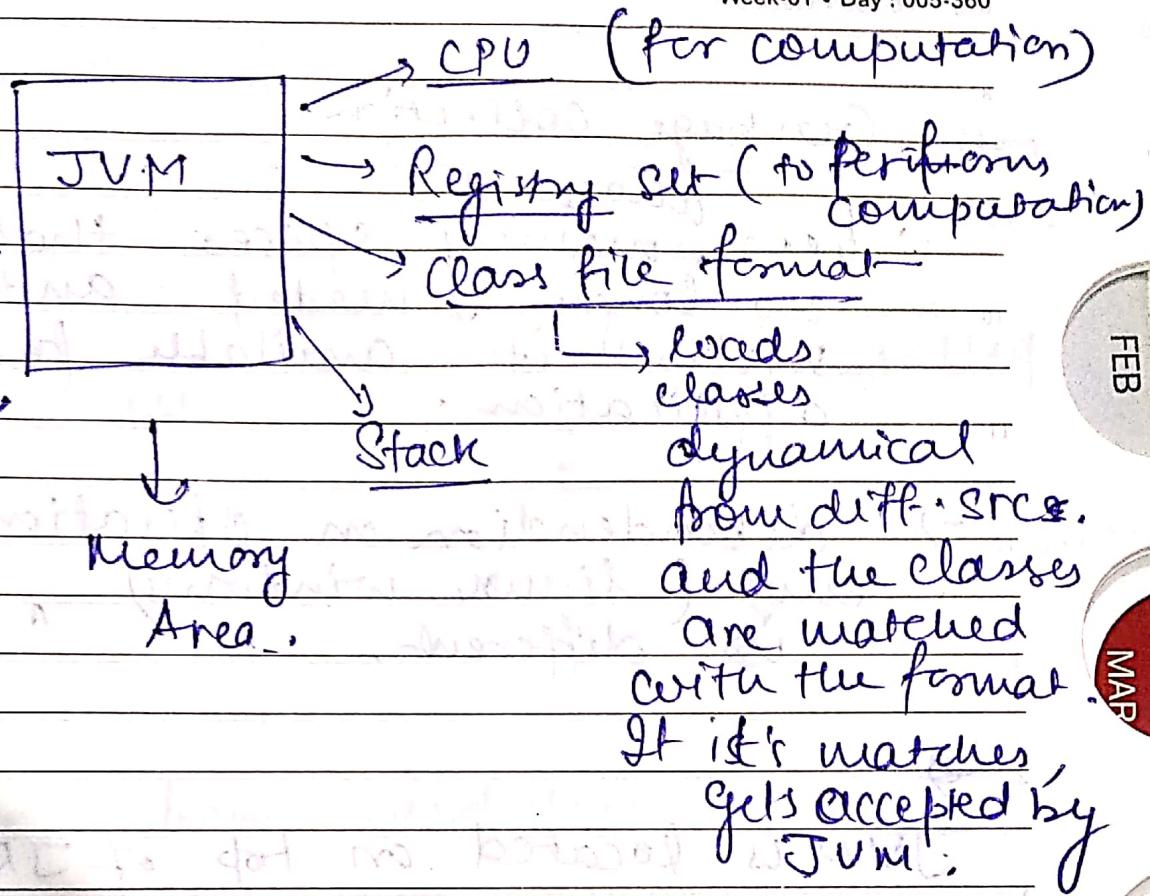
3 (1) Java Virtual Machine (JVM)

- 4 Kind of virtual envt. platform.
- 5 It provide virtualized hardware specification.

6 It can exist on your system or web browser.

Evening
7 It is a SW but implemented as h/w.

J	A	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T						
N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31



Type checking

→ Whether code is going to do memory leakage.

→ Does it adhere rules of JVM.

Evening → It must adhere compliance prescribed by sun micro system.

2	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F
0																	
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

for every OS, JVM is different.

06

January

Friday

2017

Day : 006-359 • Week-01

Garbage Collection

allocated

→ frees memory spaces that is no longer needed and makes it available for next application.

→ dependence on operational envt (linux windows), deallocation is different.

JVM is located on top of JRE.

JVM → executes program.

JRE → creates envt. to execute program.

.java → extension.

Evening

Attribute name starts with small letter
Classname always start with Cap letter

January

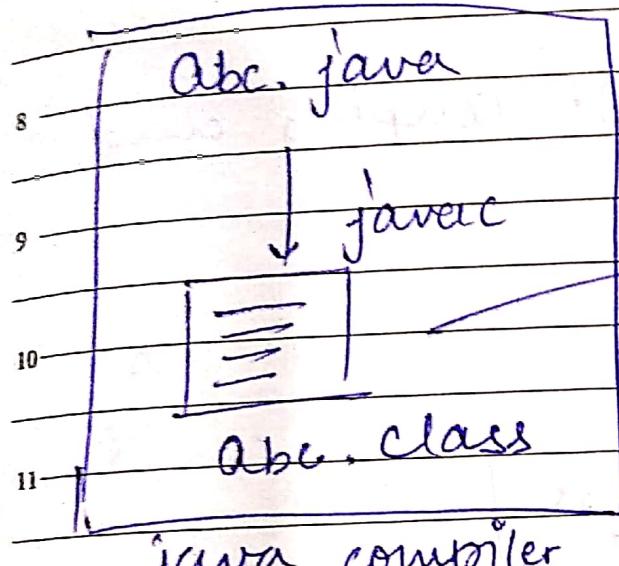
07

2017

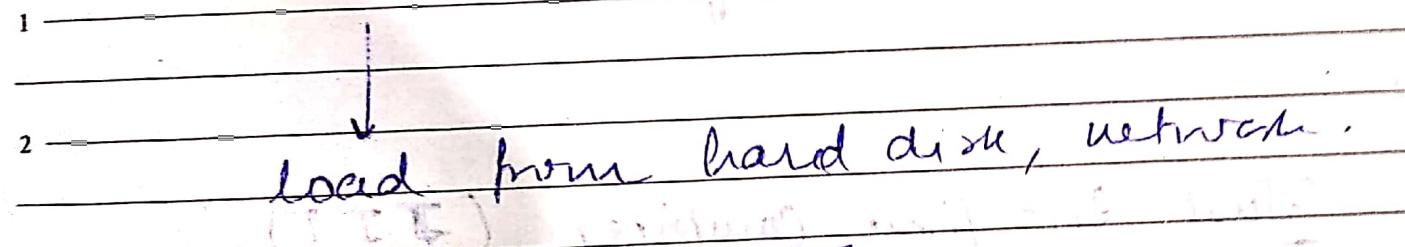
class and file name
should be same.

Saturday

Week-01 • Day : 007-358



1st time compilation
generate by te code.
can be run on
any jre installed
system. It
makes java a
platform independent



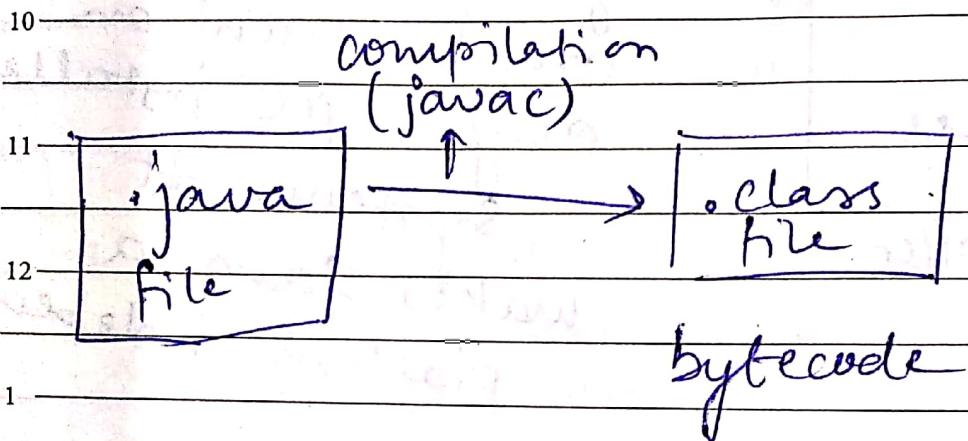
Runtime.

load necessary
classes either
remotely

Sunday 08

F

8 hybrid compilation: It compiles and
9 interprets.



3 Just In-Time Compiler (JIT)

4 Apart from calling interpreter
5 JVM calls JIT, that compiles the
code again.

6 Again and again we are
7 not doing interpretation.
Evening Execution will be faster.

J

A S M T W T F S S M T W T F S S M T W T F

N 1 2 3 4

2017

javac - HelloWorld (~ January

java HelloWord (Sun Tuesday

java
file)

10

Week-02 • Day : 010-355

Framework / API

Predefined set of rules to achieve certain problem statement.

② (solution)

Libraries are within API's, Those libraries are like Objects that are designed in such a way that they can be designed multiple times in multiple situation to solve multiple problems.

Package : Structure of

Putting all related classes in same location.

Evening

11

January

Wednesday

2017

Day : 011-354 • Week-02

8 declaring Java tech. class

9

10 $\{ \text{Modifier} \}^* \text{ class } \{ \text{Class_name} \} \}$

11 $\{ \text{Modifier} \}^* \text{ Attribute_declaration} \}$

12 $\{ \text{Constructor_declaration} \}^*$

13 $\{ \text{Method_declaration} \}^*$

2 Eg

3 public class Vehicle { }

4 private double maxload;

5 public void setMaxLoad(double value) {
6 maxload = value; }

7 }

Evening }

Method always start with small letter. January

2017 1 Feb 2022

DAY-2 Thursday

12

Week-02 • Day : 012-353

Declaring method.

Method: <return-type> <name>
(<arguments>) {

public void setWeight (int newWeight) {
if (newWeight >= 0)
weight = newWeight;

return-type is mandatory, if a method has no return type, put void. Argument is optional.
if we don't have any argument.
Java has not void as argument.

Accessing object member

dot notation.

Object > .(member)

d.setWeight = (42);

We have access to public member.

20	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	F	E
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
7	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

13

January

Friday

con. examples

I model

L, class Cjawa, Employee

2017

Day : 013-352 • Week-02

8 Encapsulation,

9 → Part of abstraction,

10 → hides the implementation details
of class.

11 Variables

instances! The no. of variables we
are creating would be a part of
particular instance.

2
3 # constructor: whether you define
or not, one constructor is
4 available i.e., default constructor.

5 It is a constructor which is
initialise by class to give
6 default values.

Evening

J

A

S

M

T

W

F

S

S

M

T

W

F

S

S

M

T

W

F

S

S

M

T

W

F

S

S

M

T

W

F

S

S

M

T

W

F

S

S

M

T

2

0

1

Multi

Week-02 • Day : 014-351

~~single~~ line comment

1 * comment

*

*

single

~~multiple~~ line comment

11 comment on line

*

1 / ** documentation comment

2 -> * Javadoc comment

*

4 javac -d, * java
5 javadoc *, *

Open index.htmle

Sunday 15

9 → eol. (end of line)

{

→ to enclose multiline with braces

W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	
2 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	B

Primitive Data type

8 - primitive -

logical : boolean .

Textual : char .

Integral : byte, short, int, long .

floating : double, float -

boolean : → true

→ false .

character : 16-bit unicode

character .

→ put { } for next class

2017

Tuesday

Week-03 • Day : 017-348

'\u00d7? ? ? ? ? '

\u2192 unicode .

\u03a6 \u2192 unique hexa code
decimal .

String; is not a primitive datatype
but a class.

" " "

\n \u2192 new line character .

8 bits byte . -2^7 to $2^7 - 1$

16 bits short -2^{15} to $2^{15} - 1$

32 bits int -2^{31} to $2^{31} - 1$

64 bits long -2^{63} to $2^{63} - 1$

2	0	1	W	T	F	S	S	M	T	W	F	S	S	M	T	W	T	F
7	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
8																		

18

January

Wednesday

2017

Day : 010-347 • Week-03

1 byte = 8 bits.

8 floating

82 bits \rightarrow float -> 4 bytes.64 bits \rightarrow double \rightarrow 8 bytes.

Reference type

reference type datatype: any object
 referring to a class.
 they are not datatype.

Class Mydate () {

Mydate today;

reference data type.

J	A	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T						
N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

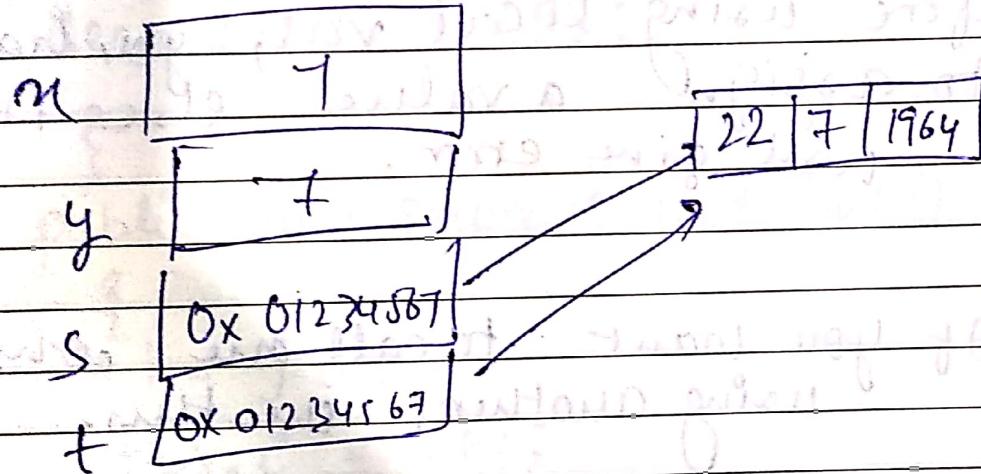
8. Two var refer to single object.

9. `int x = 7;`

10. `int y = &x;`

11. `MyDate s = new MyDate(22, 7, 1964);`

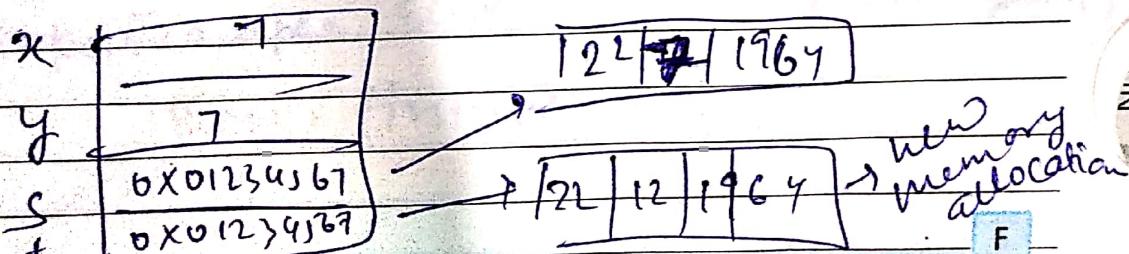
12. `MyDate t = s;`



5. Relassignment makes two variables
pts. to two object:

6. `t = new MyDate(22, 12, 1964);`

Evening



W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Day : 020-345 • Week-03

~~Call~~~~# Pass by value.~~

variable defined in class → instance variable.

(update 99) variable in method → local variable.
/automatic/temporal stack.

Before using local var, you have to assign a value else it will give error.

If you want to call one constructor using another, use this.

~~# Constructor chaining~~~~this ()~~

Evening

684460

2017

January

Saturday

21

Week-03 • Day : 021-344

Variable Scope Example

```
public class ScopeExample {
```

```
    private int i = 1;
```

```
    public void firstMethod() {
```

```
        int i = 4; j = 5;
```

```
        this.i = i + j;
```

Second Method (7);

}

```
    public void secondMethod( int i ) {
```

}

```
        int j = 8;
```

```
        this.i = i + j;
```

}

}

```
public class TestScoping {
```

Sunday 22

```
public static void main( String[] args ) {
```

```
    ScopeExample scope = new ScopeExample();
```

```
    scope.firstMethod();
```

}

20	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

7 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 . . . B

FEB

MAR

APR

JUN

1. How to take user input?
2. Create an array of Employees.
3. employeeID
employeeName
employeeSalary
4. display of Employees.

bound: If we don't put bound, we can get negative values but if we put bound, we will get the values.

Eg: bound: 100. ranging from 0 to 100

Evening

J	A	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T
N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

2017

January

Tuesday

24

Week-04 • Day : 024-341

Access Control

Modifier.	Same class	Same package	Subclass	Chiverse	Global
private	Yes				
Default	Yes	Yes			
Protected	Yes	Yes	Yes		
Public	Yes	Yes	Yes	Yes	

3. Multiple inheritance is not done in java. We can achieve multiple inheritance through interfaces.

4. Constructors can't be inherited.

5. Super() → to call base class constructor.
6. override methods.

Dynamic Method Dispatching

Evening Reference variable of base type can point ~~refer~~ to object of derived type.

Method Overloading

Same method name but different no. of arguments and types of argument.

Depending upon no. of arg and type of arg, we can overload a method.

The method can't be overloaded based on returntype (double, int).

add(int ... a) → This means it's considered ~~not~~ an array
But the exact length of this array.

∴ its ~~length~~ length is not fixed,
; its ~~is~~ a dynamic array.

Evening

J

A

S

M

T

W

F

S

S

M

T

W

F

S

javac -d . MyApp.java

java p -c com.example.demo.26app

2017

January

Thursday

26

3-Feb-2022

Day 4

Week-04 • Day : 026-339

#

Objects of class

Any class in java by default extends from the class ~~Object~~ Object.

Object class is having certain methods which we can override

12

provides the functionality

Wrapper class → if we wish to work with attribute.

It's a class which is an object representation of all primitive datatype. (boolean, char, byte, int, long, float, double, short)

by public static void main (String[] args) {

// primitive type

int a=100;

Integer x = new Integer (value: 100)
or

Integer x = new Integer (a)

(here primitive type is converted to wrapper type is k/a Boxing).

Evening

F

8 `Integer n = a;`
 9 (automatic conversion to primitive
 10 to wrapper without using new
 operator (i.e. automatic boxing)).

11 `Integer x = 100;`

12 // Unboxing.

2 `int result = x. intValue();`

3 primitive wrapper is converted
 4 to primitive type and
 5 getting assigned to
 result.

6 // auto unboxing.

7 `int result = x;`

Evening

J

A S M T W T E

final Keyword

Week-04 • Day : 028-337

If you define a class as final class, you can't inherit from that class.

If in a base class, any method is final, you can't override.

We can also define var as final.

If a var is final, its value can't be changed. It's like a constant.

Name convention of final

variable is ALL-CAPS.

Usually final variable are defined as static.

Sunday 29

free set or in-

from set.

2	0	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

F

Abstract Class

Purpose of abstract class is to design our classes. Any class which extends from abstract class ~~to~~ needs to override the abstract method.

By default all methods in interface are abstract.

Abstract class can have non-abstract method. but interface can't & can't have ~~be~~ non-abstract methods. Better to use interface.

2017

Interfaces.

January

Tuesday

31

Week-05 • Day : 031-334

A class can implement more than 1 interfaces but can't extends more than 1 class.

Static variables can directly be accessed by using . operator.
(MyVehicle.str)

Interface can't contain any private member.

public static final String str = "Hello";

4. Feb. 2022. • Day 5

2017

Notes

Exception Handling.

Exception: Unexpected situation caused due to illegal argument, file not found, network failure.

Assertion: By this we can check whether the application is working as per expectation or not.

Throwable
Class

Error
Class

Exception
class

Exception can be handled but error can't.

∴ Error is unchecked (memory failure, file failure or crashes).

∴ Exception is checked (they can be handled. If there is no handler, then JVM handles the exception).

201

Week 05 * Day 1 632-333

The code which might create exception are kept in try-block.

QUESTION

Logic

$$\text{age } (\geq 00 \text{ } \cancel{\leq} \text{ } 45) \geq$$

use string for contact number.
make sure contact number is 10 dig.

take string \rightarrow convert to integer.

Output

id:

Name: Mr. Siddhartha Bhattacharya
address: 12, Sector 1, Salt Lake, Kolkata - 700061

Mobile No.: 9833333333

Evening

	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F
2															
0															
1	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F

M
A
R

卷之三

Regulation

- 2000-2001
Year 1
Year 2
Year 3
Year 4
Year 5

Thimble

Exceptions.
(checked)

10. Null pointer exception (Arithmetic Exception | Null pointer Exception)

- 12
1) Arithmetic
2) Macro pointer
3) Illegal dg.

Finally: Whether exception occurred or not, finally block is going to execute everytime.

Evening

卷之三

七

Collection Framework

Day: 035-330 • Week: 05

- 8 Collection is a single object which holds another object.
- 9 ~~for a collection we share objects.~~
- 10 ~~for a collection we share objects.~~

- 11 → light-weight desp. with POJO
- 12 Spring is a framework.

- 13 → light-weight desp. with POJO
- 14 Spring old gave better.

- 15 →

- 16 → light-weight desp. with POJO
- 17 → light-weight desp. with POJO

- 18 → light-weight desp. with POJO
- 19 → light-weight desp. with POJO

- 20 → light-weight desp. with POJO
- 21 → light-weight desp. with POJO

- 22 → light-weight desp. with POJO
- 23 → light-weight desp. with POJO

- 24 → light-weight desp. with POJO
- 25 → light-weight desp. with POJO

- 26 → light-weight desp. with POJO
- 27 → light-weight desp. with POJO

- 28 → light-weight desp. with POJO
- 29 → light-weight desp. with POJO

- 30 → light-weight desp. with POJO
- 31 → light-weight desp. with POJO

- 32 → light-weight desp. with POJO
- 33 → light-weight desp. with POJO

- 34 → light-weight desp. with POJO
- 35 → light-weight desp. with POJO

- 36 → light-weight desp. with POJO
- 37 → light-weight desp. with POJO

- 38 → light-weight desp. with POJO
- 39 → light-weight desp. with POJO

Spring framework

Week: 06 • Day: 037-328

- 8 → framework: predefined set for a problem.
- 9 →

- 10 → Spring is a framework.

- 11 → light-weight desp. with POJO
- 12 → light-weight desp. with POJO

- 13 → light-weight desp. with POJO
- 14 → light-weight desp. with POJO

- 15 → light-weight desp. with POJO
- 16 → light-weight desp. with POJO

- 17 → light-weight desp. with POJO
- 18 → light-weight desp. with POJO

- 19 → light-weight desp. with POJO
- 20 → light-weight desp. with POJO

- 21 → light-weight desp. with POJO
- 22 → light-weight desp. with POJO

- 23 → light-weight desp. with POJO
- 24 → light-weight desp. with POJO

- 25 → light-weight desp. with POJO
- 26 → light-weight desp. with POJO

- 27 → light-weight desp. with POJO
- 28 → light-weight desp. with POJO

- 29 → light-weight desp. with POJO
- 30 → light-weight desp. with POJO

- 31 → light-weight desp. with POJO
- 32 → light-weight desp. with POJO

- 33 → light-weight desp. with POJO
- 34 → light-weight desp. with POJO

- 35 → light-weight desp. with POJO
- 36 → light-weight desp. with POJO

- 37 → light-weight desp. with POJO
- 38 → light-weight desp. with POJO

- 39 → light-weight desp. with POJO
- 40 → light-weight desp. with POJO

F	E	M	T	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

07

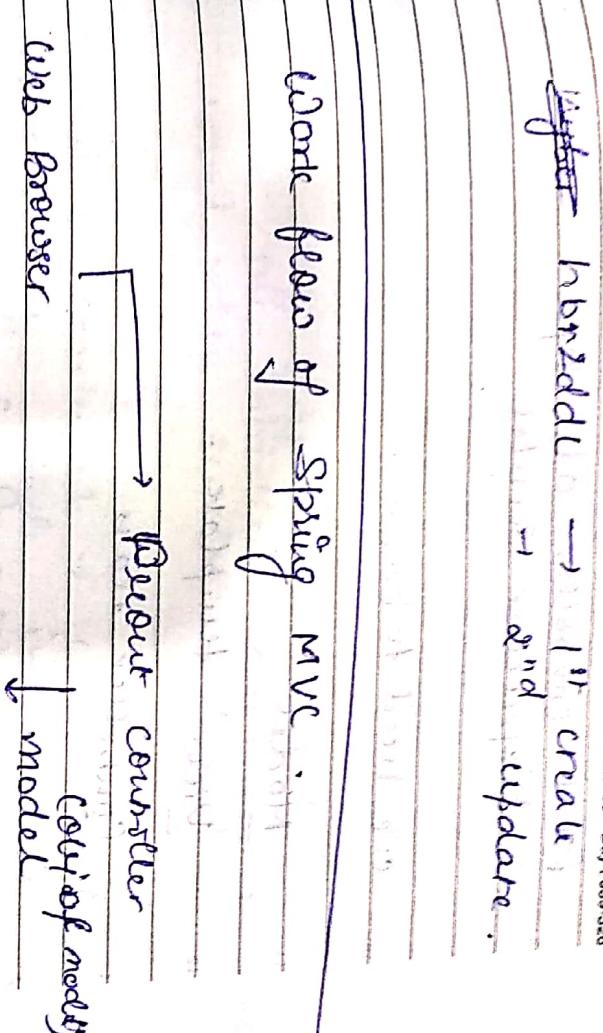
Tuesday

Hibernate

- 1 Date class - week 08
- 2 It's a framework, where we use this class for persisting Java objects in database.
- 3 We are going to store Java object in memory from statements.
- 4 It's an object relational mapping tool.
- 5 Session
- 6 → Add hibernate config file.
- 7 → Annotations from class.
- 8 dialect: top layer on jdbc used by the driver.
- 9 used with entity.
- 10 @ Data is used with Entity
- 11 getter
- 12 setter
- 13 To String
- 14 hashCode.

Week 08 • Day 039 328

08



Front controller also known as
Dispatcher Service

we need to create -

Model

View template .

Controller class

2

controller! Code created by developer

store data and place

data to model and

handles request .

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

A

F	E	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	
2	1	0	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	
3	2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	R

Web.xml → Also known as deployment descriptor.

init-param → initialized parameter.

when my dispatcher servlet is loaded, then init-param initialized attribute is used when servlet is loaded.

↓ located on startup → when it is loaded then only single instance is created for servlet.

↓ transient (Annotation) : If we are declaring any variable with @transient annotation, then in SQL, no column of such variable will get generated.

↓ that variable is only going to perform business logic.

↓ transient is non-persistent layer.

Evening

Day : 042-323 • Week-06

8 @Query: annotation to declare finder
9 query directly by repository layer.

10 

11 # Resilience 4j (circuit breaking pattern)

12

1

2

3

February

2017

February

Day: 047-318 • Week-07

17

Friday

React JS → Framework

Week-07 • Day: 048-317

DOM

Document Object Model

→ whole html ~~content~~ document is loaded

in browser, it becomes document object.

Imported methods of document object

write ("string") → writes given string on document

writeln ("string") → " " on new line

getElementById ('')

getElementByName()

getElementsByName()

getElementsByTagName()

getElementsByTagNameByClassName()

→ It is deployed in tree hierarchy.

Evening

F	E	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F	S	S	M	T	W
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

F	E	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F	S	S	M	T	W
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

F	E	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F	S	S	M	T	W
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

F	E	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F	S	S	M	T	W
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

F	E	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F	S	S	M	T	W
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

F	E	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F	S	S	M	T	W
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

F	E	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F	S	S	M	T	W
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

F	E	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F	S	S	M	T	W
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

F	E	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F	S	S	M	T	W
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

F	E	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F	S	S	M	T	W
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
R	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

~~metoo~~ 18 February 2017
Node -v (node version)
npm -v (npm version)

Day : 049-316 • Week-07

2017

Monday

20

Year 2016 • Day 049/365

React components

Class

functional
(Bands from React
properties as input
component and
function)

and returns what should
be rendered

npm install react react-dom

npm install -g create-

react-aapp

(install react and its
dependencies)

create-react-app & pt-

npm start
(run react-app)

→ cd react-aapp
→ npm start

React & state
(useState)
getInitialstate() → render()
(to define
a state, add
useState
thus, state
render) memo

→ cd react-aapp
→ npm start

→ npm start

19 Sunday

Evening

F E M T W T F S S M T W T F S S
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

1

21

February props → properties and
read-only
component

2017

2017

February

Wednesday

22

Day : 052-313 • Week-08

Week-08 • Day : 053-312

React Component Life Cycle

- Initial phase → get default props()

- get initial state()

- Mounting phase → component will mount

- componentDidMount()

- render()

- Updating phase → component will receive props

- Should componentUpdate() → componentDidUpdate()

- Unmounting phase → componentWillUnmount()

React Create Project

npx create-react-app myfirst-react-app

- extended npx

npm → starts on 3000

- start

- 6

- 7

SWAGGER

2017

open api documentation dependency.

December

Monday

18

Week-01 • Day : 352-013

It's going to do automatic documentation of rest api.

Swagger :

open API definition.
It detects ur ~~endp~~ application endpoint.

localhost:8081/swagger-ui.html.

Do in controller class

@operation (summary = "Employee as dependency") open UI.

11 Optional : If ~~is~~ method has no
12 ~~return~~ value to return, then
by optional, no need to
~~do~~ exceptional handling.
It will by default throw
null.

Angular Routing → deployed through angular
close n restart
install: wsl2

- 11 one installation is completed etc ⇒
- 12 docker run hello-world on cloud

Angular Project

cd my-first-angular-project /
install bootstrap in Angular
install

→ npm i bootstrap

→ we require bootstrap.min.css

- 1 → style.css → we can add global style
- 2 → @import "node_modules/bootstrap/dist/css/bootstrap.css";

then use bootstrap in html file.

- 3 → generate
- 4 → ng g c todo-service
- 5 → ts component
- 6 → ng generate class todo-service/Todo
- 7 → ng generate class Todo

New task
todo id
to-do name

to-do name is duplicate

14 Tuesday

Day : 346-019 • Week-50

→ Central repository for docker.

hub.docker.com

→ Eg: docker pull mysql.

it will bring image from remote registry to local system.

(consider local system doesn't have mysql).

→ docker images.

→ docker container ls -a

To run this img as container.

docker container run -it -e name

If we don't give name to container,
it will take arbitrary name.

port ↗
8307: 8306 ↗ by default

→ docker port -

→ blocked in local system is b

exposing / mapping port
Evening
to 8307

8 Docker

9 sets up env. that take care
of shipping of applications.

10 → virtual representation of container.

11 → Images contain code, OS, mysql,
torrent server.

1 → dev. app.

2 → ship it in container containing image.

3 → If we wish to

4 → Img can't run, container runs.

5 → If img has mysql.

6 → It will look to mysql container.

7 → If it doesn't get mysql container
it will search for

Evening

December

2017

Monday

11

Build docker tomcat

Week-50 • Day : 345-020

8 docker build -t helloworld : latest .

9 docker images

10 docker container

11 docker container ls -a

12 docker system prune -a .

13 docker container run -p 8081:8080
helloworld : latest .

1 Push project on docker

2 → docker login

3 → docker images

4 → docker push akash121299/emp

5 ↑
image name

6 detach mode : we will not get
7 console of log.

(how x11 is implemented, etc.)

Evening