* Type of constructors :-

- There are two types of constructors in Java:

1. Default constructor (no-arg constructor)
2. parameterized constructor

If there is no constructor in a class, compiler automatically creates a default constructor.

Types of Java constructor:-

1. Default constructor.
2. parameterized constructor.

* Constructor chaining

- calling a constructor from the another constructor of same class is known as constructor chaining.

* This and super constructors

- this() and super() are used to call constructors explicitly.

1. Using this() you can call the current class's constructor

2. Using super() you can call the constructor of the super class.

* Exception :-

- What? Why? How?

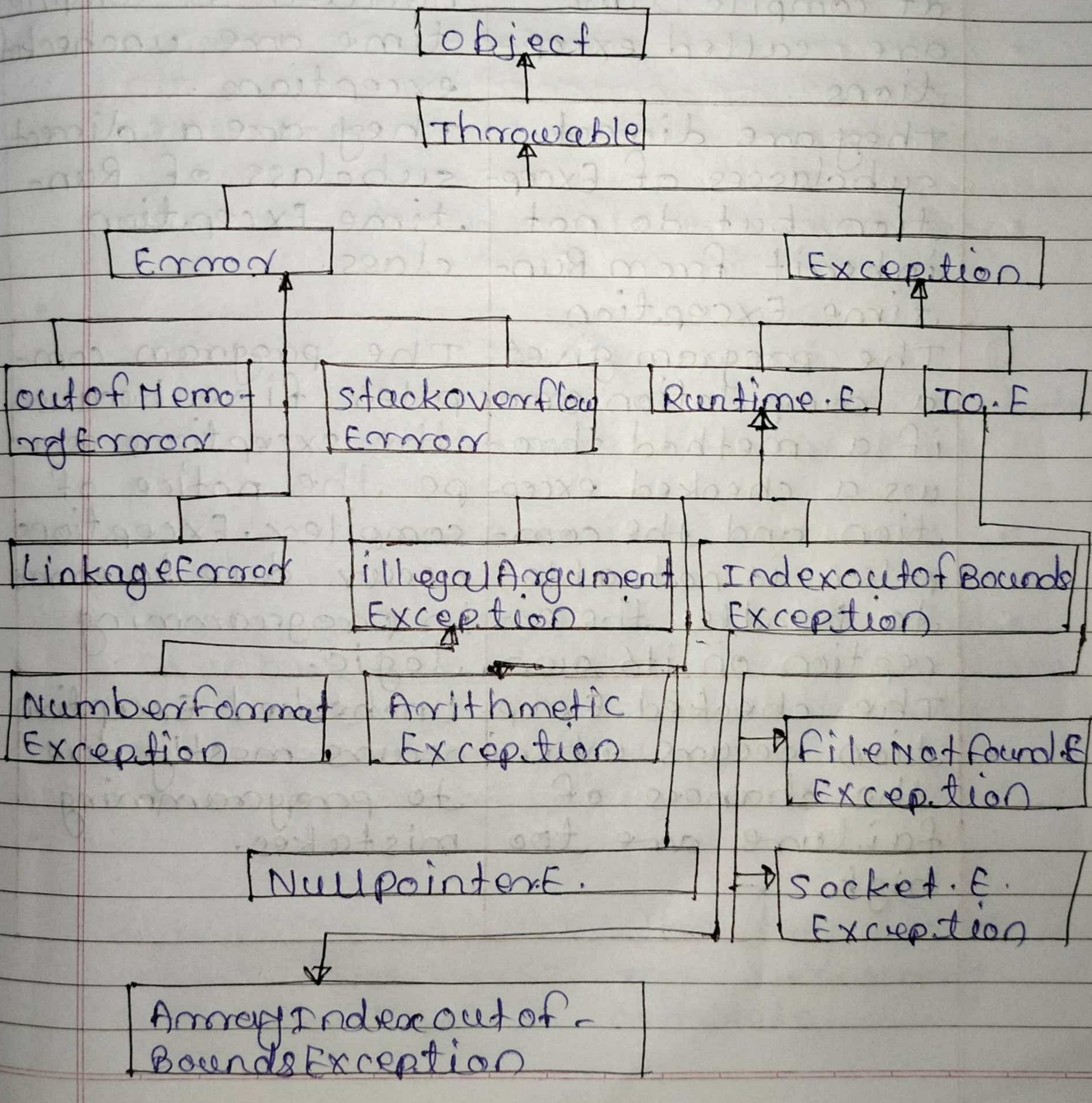
- In Java, an exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

Exception Handling is a mechanism to handle runtime errors such as classNotFoundException, IOException, SQLException, RemoteException, etc.

The core advantage of exception handling is to maintain the normal flow of the application.

An exception normally disrupts the normal flow of the application that is why we use exception handling.

* API hierarchy for Exceptions



Page No.
Date

* * Checked Exception
Vs
Unchecked Exception *

Exception that are checked and handled at compile time are called exceptions.	Exceptions that are not checked and handled at compile time are unchecked exceptions.
They are direct subclasses of Exception but do not inherit from Runtime Exception.	They are a direct subclass of Runtime Exception class.
The program gives a compilation error if a method throws a checked exception and the compiler is not able to handle the exception on its own.	The program compiles fine because the exceptions escape the notice of compiler. Exceptions occur due to errors in programming logic.
The checked Exception occurs when the chances of failure are too high.	Unchecked Exception occurs mostly due to programming mistakes.

* Keywords for Java Exceptions

- Throws

  Describes the exceptions which can be raised by a method

- Throw

  Raises an exception to the first available handler in the call stack, unwinding the stack along the way.

- Try

  Marks the start of a block associated with a set of exception handlers.

- catch

  If the block enclosed by the try generates an exception of this type, control moves here; watch out for implicit subsumption.

- Finally

  Always called when the try block concludes, and after any necessary catch handler is complete.

Page No.
Date

* string in Java :-

* strings :-

- string is a sequence of characters placed in double quote (" "). // ("Java")

- strings are constant, their values cannot be changed in the same memory after they are created.

- There are two ways to create string object :

1) By string literal.
2) By new keyword

* string creation :-

- By string literal :

For Example : string s1 = "welcome";
string s2 = "welcome";
//no new object will be created

Page No.
Date

- By new keyword:

For Example:

```
string s = new string ("sachin");
string s = newstring ("sachinTendulkar");
// create two objects and one reference
variable.
```

* Immulability :-

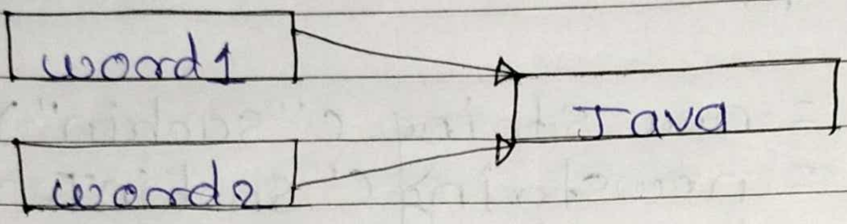
- In java, string objects are immutable. Immutable simply means unmodifiable or unchangeable.

once string object is created its data or state can't be changed but a new string object is created.

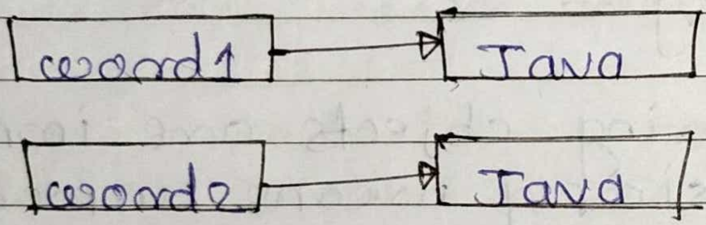
Advantage of immutability is use of less memory.

Disadvantages of immulability : Less efficient - you need to create a new string and throw away the old one even for small changes.

string word1 = "Java"
string word2 = word1;



string word1 = "Java";
string word2 = new string (word1);



Less efficient : wastes memory

* string Methods :

- substring (int begin):-

Returns substring from begin index to end of the string.

Example:- string s = "narce";
system.out.println (s.substring(2));
//rce

- equals ():-

To perform content comparision where case is important.

Example :- string s = "Java";
system. out. println (s. equals ("java"));
//true

- concat ():-

Adding two strings we use this method

Example :- string s = "nacre";
s = s. concat ("software");
system. out. println (s); // nacresoftware

- length (), charAt ()

int length (); Returns the number of characters in the string.

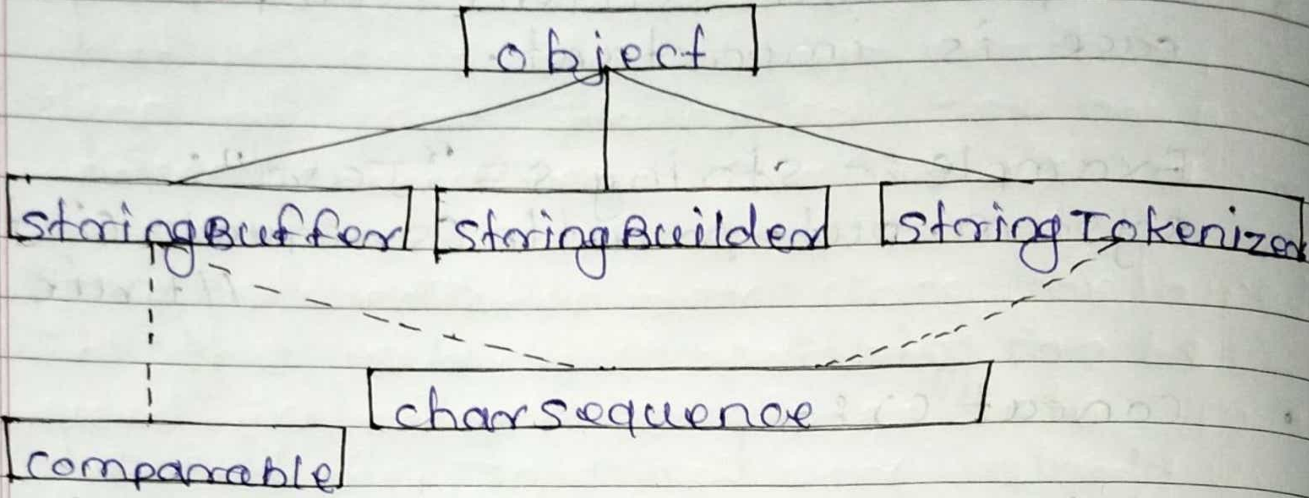
char charAt (i); Returns the char at position i.

character positions in strings are numbered starting from 0 - just like Arrays.

Returns:
"problem". length (); ⟶ 7
"Window". charAt (2); ⟶ 'n'

Page No.
Date

* stringBuffer, stringBuilder and stringTokenizer



All these classes are final and subclass of serializable.

* Limitation of string in Java?

What is mutable string?

- A string that can be modified or changed is known as mutable string. stringBuffer and stringBuilder classes are used for creating mutable string.

- Difference between string and string-Buffer in java?

Main difference between string and stringBuffer is string is immutable while stringBuffer is mutable.

Page No. | Date

* StringBuffer :-

- StringBuffer is a synchronized and allows us to mutate the string.

- StringBuffer has many utility methods to manipulate the string.

- This is more useful when using in multi-threaded environment.

- Always modified in same memory location.

- Methods :-

1) Append 2) Insert
3) Delete 4) Reverse
5) Replacing character at given index.

* StringBuilder :-

- StringBuilder is the same as the String-Buffer class.

- The StringBuilder class is not synchronized and hence in a single threaded environment, the overhead is less than using a StringBuffer.

Page No.
Date

* stringTokenizer

- A token is a portion of a string that is separated from another portion of that string by one or more chosen characters (called delimiters).

- The stringTokenizer class contained in the java. util ·package can be used to break a string into separate tokens. This is particularly useful in those situations in which we want to read and process one token at a time ; the BufferReader class does not have a method to read one token at a time.

Index	string	stringBuffer	string Builder
storage Area	constant string pool	Heap	Heap
Modifiable	No (immutable)	yes (mutable)	yes (mutable)
Thread safe	yes	yes	NO
Thread safe	Fast	very slow	fast

Page No.
Date

* StringBuffer vs StringBuilder

StringBuffer	StringBuilder
1. Thread-safe	1. Not Thread-safe
2. synchronized	2. Not synchronized
3. since Java 1.0	3. since Java 1.5
4. slower	4. Faster.

THANK YOU

*