**Day 7: 24-10-2025:**

**String :** String is a pre defined or also known as reference data types

Creating syntax for String class object

String str1 = “Welcome to Java”;

String str2 = new String(“Welcome to Java”);

== : it check contents and memory code

equals : it check only contents doesn’t matter same memory or different memory.

equals() method is part of Object class. String class as well as all Wrapper classes internally overrided equals method to check to object value and their memory code

String is known as immutable class. means we can’t change the value.

StringBuffer and StringBuilder : these two classes are type String mutable classes.

StringBuffer by default methods are synchronized. So it is thread safe but slow in performance in multithreading environment.

StringBuilder : not a synchronized so not thread safe. But fast in performance.

All these 3 classes and final. So we can’t inherits those classes.

**Generics**

Generics is way by which programmer can specify the type of object that class can work with Via parameter passes at declaration time and evaluate at compile as well as run time.

Non generic code in Collection framework

List ll = new ArrayList();

ll.add(10);

ll.add(10.20);

ll.add(“ravi”);

Object obj = ll.get(0);

integer i = (Integer)obj;

int n = i.intValue();

Generic code in Collection framework

List<Integer> ll = new ArrayList<>();

ll.add(10);

ll.add(20);

ll.add(30);

int n = ll.get(0);

Generic class : The generic class is a mechanism to specific the type of relationship between the component type and its object.

The syntax of declaration of generic class as normal class with angular brackets. The type of parameter are declared.

E -🡪 Element type

K 🡪 Key type

N 🡪 Number type

T 🡪 return type

V 🡪 value type

class Test<T> {

Test(T obj) {

System.out.println(obj);

}

}

Test t1 = new Test(10);

Test t2 = new Test(10.20);

Test t3 = new Test(“Ravi”);

If we create generic class with Type as T or any alphabets that T we can use inside constructor as well as methods.

If we need type for particular method. Then you can use generic method.

**Generics wild card**

In Generics we use the wild card as ? it represents an unknown types. This wild card we can use in variety of situation as a type of parameter, fields, local variable as well as return type.

1. ? any type
2. ? extends Type (upper bond)
3. ? super Type (lower bond)

**Multi threading :**

Program : set of instruction to perform any specific task.

Process : time taken to execute the code or program in execution.

Processor : process is responsible to execute the code or process the code.

Thread : thread is small execution of a code within process.

Thread also known as light weighted process means it takes less resource or memory to do same task which we do using processor.

Java by default multi threading programming language.

In side main method always one default thread execute without our knowledge.

currentThread(); this method is static method part of Thread class and this class is part of lang package. method return type is Thread class reference.

Thread t = Thread.currentThread();

t 🡪 it will provide the information about default thread ie thread name, thread priority and thread group

by default thread name is main, priority is 5 and group is main.

Priority

Min 🡪 1

Norm 🡪 5

Max 🡪 10

Multi tasking : more than one task.

1. Process base
2. Thread base

Creating more than one thread in java

1. Extends Thread class
2. Implements Runnable interface
3. Implements Callable interface

Extends Thread class.

1. We need to create user defined class and that class must be extends Thread class.
2. Then create thread class (ie the class which extends Thread class also known as thread class).
3. Using thread class reference call start() method. start() is a pre defined method which help to start the thread is Runnable state(Ready to run).
4. Start() method internally call run() method. by default run() method part of thread class contains empty method body. If you want to execute any custom code then you need override run the method and provide logic which you want to execute.

Implements Runnable

1. Create user defined class and that class must be implements Runnable interface.
2. When class implements Runnable interface you need to override run method mandatory. Run() is a pre defined method part of Runnable interface and Thread class internally implements Runnable and provided empty body for run() methods.
3. Create the thread class reference and call start() method.
4. Create the object of thread class using new keyword and pass the reference of Runnable through constructor.
5. With help of reference call start(), which internally call run() method. insider run() method provide custom logic to perform.

Life cycle of the thread

Create ---🡪 Runnable state -🡪 Running --- > Destroy

obj1 obj1.start() run i=5, i=10

t1 t1.start();

inside a run() method for user defined thread and inside main method for pre defined thread we can use few method to control the thread execution

sleep()

isAlive()

join

wait()

notify()

notifyAll()