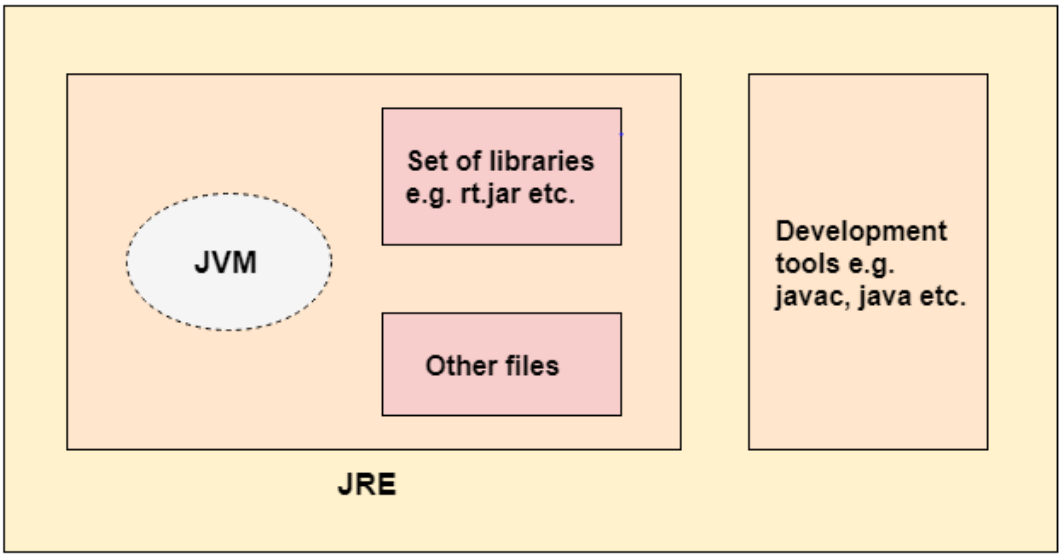

JDK vs JRE vs JVM

Day 3

Discussed about language
Java being ranked 2nd for many years.
.java<--javac->Bytecode(.class)<-->BinaryCode(Machine)
Mixed mode:Compiled+interpreted
Compile time exceptions
Java 7,8,9,10
Java 8 functional programming
JDK versions Java SE8 is a commmercial implementation by Oracle
Compiler (JDK)-1.8
RunTime (JRE) - > 1.8
Install JDK = JDK+JRE
libraries are under bin folder.
Object Oriented

```
class Test{}//execution code must remain inside class.
class Demo1{
        public static void main(String a[]){
        System.out.println("ddd");
        }
}
```

Classes -> Grouping (Package <--> Folder hierarchy)
java.lang,.util,.sql-->JavaSE
javax.-->JavaEE
org.spring.
import java.lang.System;//everything comes from here
//Filename and public class name must be same

package xyz;
public class Cal{public int add(int i,int j){
return i+j;}}

//Cal.java
javac -s. calc.java//source file
-d.//destination
//mapping package name to directory structure

```
class Demo1{
        public static void main(String a[]){
        Cal c= new Cal();
        int ans=add(10,20);
        System.out.println("ans"+ ans);
        }
}
```

//Demo1.java
//javac Demo1.java
//java Demo1

//Activity for calulator uploaded in: "https://github.com/TusharrS/apisero-assignment-activities"

//oracle documentation for Datatypes:
"(https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html)
"
//Scanner class

//Activity: "https://github.com/TusharrS/apisero-assignment-activities"

//sysout CTRL+Space
//Scanner CTRL+Space//to import library
//Java Naming conventions from oracle documenttion
//Right cllck ->Source ->Format
//avg. of 5 numbers
int[] arr=new int[5];//integer array
Arrays.toString(arr[]) function
//.length//function for int array
//arr of str and avg. length of name

//to String --> "ClassName@HashCode "

//@Override method
//getter-setter
//print(int i,int i){
System.out.print(+i+""+j);
i+=10;
j+=10;
Print
}

//call by value and call by reference
//Exercisesbased on getter,setter,constructor
//access modifiers scope

Access Modifier	within class	within package	outside package by subclass only	outside package
Private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y

//inheritance together with access modifiers

class Subclass-name extends Superclass-name
{
//methods and fields

//overloading

```
class OverloadingExample{

static int add(int a,int b){return a+b;}
static int add(int a,int b,int c){return a+b+c;}
}
```

//Overriding in classes

```
class Animal{

void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void eat(){System.out.println("eating bread...");}  }
```

Day4

Overview of Day3
Class -> abstract class
create a class -> 1. Inheritance 2. Static
Can't create instance of class
Can be extended
Can contain normal and abstract method.

2. Class as Final
Create a class --> 1. only use it
Can create instance of that class
Can't be used as parent class
Final method cannot be overriden

One java class can't extend mutiple java classes

3. Interfaces
Implementing interface and restrictions based on instantation.
Pure abstrct class

```
abstract class Bike{
 abstract void run();
}
class Honda4 extends Bike{
void run(){System.out.println("running safely");}
public static void main(String args[]){
 Bike obj = new Honda4();
 obj.run();
}
}
```

no actual methods,variables,only abstract methods
implemnts interface
write code for all interface methods

```
interface <interface_name>{

  // declare constant fields
  // declare methods that abstract
  // by default.
}
```

4. Exception Handeling --try..catch..

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

```
  try{
    //code that may raise exception
    int data=100/0;
  }catch(ArithmeticException e){System.out.println(e);}
  //rest code of the program
  System.out.println("rest of the code...");
 }
}
```

5. Wrapper Class:Integer discussion ,etc.

---

6. Collection Framework

--> As a interface

List:Collection of items,no unique check .no ordering

Set

Map

java.util.ArrayList//for new ArrayList();

convert int to Integer in ArrrayList//Boxing

TreeSet for ordering in ascending order and to remove duplicacy

HashSet:No ordering but removes duplicacy

//Worked with LinkedList and ArraList,HasSet and TreeSet

Key Value (Directory)

Map: TreeMap,HashMap

```
Set<Integer> l1 =new HashSet<Integer>();
                    l1.add(2);
                    l1.add(12);
                    l1.add(2);
                    System.out.println(l1);

                    Map<String,Integer> mp = new TreeMap<String,Integer>();
                    mp.put("Ram", 100);
                    mp.put("Shyam", 1000);
                    mp.put("Ram", 10);
                    System.out.println(mp);
```

---

7. Threads

Multi threaded execution

```
class Multi extends Thread{
public void run(){
System.out.println("thread is running...");
```

```
}
public static void main(String args[]){
Multi t1=new Multi();
t1.start();
 }
}
```

//Running a thread

```
class TestCallRun1 extends Thread{

 public void run(){
  System.out.println("running...");
 }
 public static void main(String args[]){
  TestCallRun1 t1=new TestCallRun1();
  t1.run();//fine, but does not start a separate call stack
 }
}
```

//Had discussions about time slicing in multithreading for example functioning in eclipse .
