FAQ's
====
1. Please make the pictorial representation of this 2D array it is very confusing.
      arr[][] = {{3,4}, {2,3,4}, {1,2,3,4,5}};

2. Local or instance method how memory allocated of String in heap or stack. If local variable memory
   allocated in stack. how is it coming in stack?

3. What is static class in Java? what is the use of static class and Why static class is required
   (means when we get to know when we should go static class).

++++++++++++++++++++++++++++
Abstract class vs interfaces
++++++++++++++++++++++++++++

abstract modifier
  => if we don't know the implementation still we can declare the method with abstract modifier.
  => if the class contains or doesn't contains abstract methods then the class can be marked as "abstract class".
  => For an abstract class,object cannot be instantiated.
  => Abstract methods are such method which have only declaration but not implementation.

what is an abstract class?
 Ans. A class with or without any abstract methods ,such type of classes are called as "abstract class".
      For an abstract class, object can't be instantiated.

eg::
```
 abstract class Test{
      public static void main(String... args){
            System.out.println("sachin");
      }
 }
```

eg1::
```
class Parent
{
      public void m1();
}
```

eg2::
```
class Parent
{
      public abstract void m1();
}
```

eg3::
```
class Parent
{
      public abstract void m1(){

      }
}
```
All the above 3 are invalid

eg4::
```
abstract class Parent
{
      public abstract void m1();
}
```

eg5::
```
abstract class Parent
{
      public void m1(){

      }
      public void m2(){

      }
}
```
 => 4 and 5 are valid.(even if the class doesn't contains any abstract methods, still the class can be marked as abstract)

Question
```
class Foo {
      public int a = 3;
      public void addFive() { a += 5; System.out.print("f "); }
}
class Bar extends Foo {
      public int a = 8;
      public void addFive() { this.a += 5; System.out.print("b " ); }
}
```
Invoked with:
```
Foo f = new Bar();
f.addFive();
System.out.println(f.a);
```

What is the result?
A. b 3
B. b 8
C. b 13
D. f 3
E. f 8
F. f 13
G. Compilation fails.
H. An exception is thrown at runtime.

Q>
```
class Vehicle{
      int x;
      Vehicle(){
            this(10);
      }
      Vehicle(int x){
            this.x = x;
      }
}
class Car extends Vehicle{
      int y;
      Car(){
            super();// line -n1
```

```
            this(20);//line -n2
      }
      Car(int y){
            this.y= y;
      }
      public String toString(){
            return super.x + " " + super.y;
      }
}
public class Test {
      public static void main(String[] args) {
            Vehicle y= new Car();
            System.out.println(y);
      }
}
```

Predict the answer

A. 10:20
B. 0:20
C. Compilation fails at line n1
D. Compilation fails at line n2

Answer: D

Q>

Given:

```
      abstract public class Employee {
                  protected abstract double getSalesAmount();
                  public double getCommision() {
                              return getSalesAmount() * 0.15;
                  }
      }
      class Sales extends Employee {
                  17. // insert method here
      }
```

Which two methods, inserted independently at line 17, correctly complete the Sales class? (Choose two.)

A. double getSalesAmount() { return 1230.45; }
B. public double getSalesAmount() { return 1230.45; }
C. private double getSalesAmount() { return 1230.45; }
D. protected double getSalesAmount() { return 1230.45; }

Answer: B,D

access modifiers

```
  private
    |
  default
    |
  protected
    |
  public
```

QUESTION

Given:

```
1. class X {
2.    X() { System.out.print(1); }
3.    X(int x) {
4.          this(); System.out.print(2);
5.    }
6. }
```

```
7. public class Y extends X {
8.    Y() { super(6); System.out.print(3); }
9.    Y(int y) {
10.         this(); System.out.println(4);
11.   }
12. public static void main(String[] a) { new Y(5); }
13.}
```

What is the result?
A. 13
B. 134
C. 1234
D. 2134
E. 2143
F. 4321

Answer: C

Given:

```
1. public class A {
2.          public void doit() {
3.          }
4.          public String doit() {
5.                return "a";
6.          }
7.          public double doit(int x) {
8.                return 1.0;
9.          }
10.}
```

What is the result?
A. An exception is thrown at runtime.
B. Compilation fails because of an error in line 7.
C. Compilation fails because of an error in line 4.
D. Compilation succeeds and no runtime errors with class A occur.

Answer: C

Q>

```
public class Base {
      public static final String FOO = "foo";
            public static void main(String[] args) {
                  Base b = new Base();
                  Sub s = new Sub();

                  System.out.print(Base.FOO);//foo
                  System.out.print(Sub.FOO);//bar
                  System.out.print(b.FOO);//foo
                  System.out.print(s.FOO);//bar
                  System.out.print(((Base) s).FOO);//foo
            }
}
class Sub extends Base {
      public static final String FOO = "bar";
}
```

What is the result?
A. foofoofoofoofoo
B. foobarfoobarbar
C. foobarfoofoofoo
D. foobarfoobarfoo

E. barbarbarbarbar
F. foofoofoobarbar
G. foofoofoobarfoo

Answer: D

Given:
```
1. class Mammal {
2. }
3.
4. class Raccoon extends Mammal { // IS -A
5.    Mammal m = new Mammal();//HAS-A
6. }
7.
8. class BabyRaccoon extends Mammal {//IS-A
9. }
10.
```

Which four statements are true? (Choose four.)
A. Raccoon is-a Mammal.
B. Raccoon has-a Mammal.
C. BabyRaccoon is-a Mammal.
D. BabyRaccoon is-a Raccoon.
E. BabyRaccoon has-a Mammal.
F. BabyRaccoon is-a BabyRaccoon.

Answer:: ABCF

Question
```
abstract class C1 {
      public C1() { System.out.print(1); }//super();
}
class C2 extends C1 {
      public C2() { System.out.print(2); }//super();
}
class C3 extends C2 {
      public C3() { System.out.println(3); }//super();
}
public class Ctest {
      public static void main(String[] a) { new C3(); }
}
```
What is the result?
A. 3
B. 23
C. 32
D. 123
E. 321
F. Compilation fails.
G. An exception is thrown at runtime.

Answer:: 123(D)

sir diff between pass by value and pass by refrence , and java why uses pass by value ?
```
 => void add(int a,int b)
    {

    }
```

```
   int a = 5;
   int b = 10;
     add(a,b);//pass by value

eg#2.
    Integer a= 10;
    Integer b= 20;
      void add(Integer x, Integer y)
      {

      }
    add(a,b);//pass by reference


Q>
Given:
1. class TestA {
2.    public void start() { System.out.println("TestA"); }
3. }
4. public class TestB extends TestA {
5.    public void start() { System.out.println("TestB"); }
6.    public static void main(String[] args) {
7.          ((TestA)new TestB()).start();
8.    }
9. }
What is the result?
A. TestA
B. TestB
C. Compilation fails.
D. An exception is thrown at runtime.

Answer: B

++++++++++
interfaces
++++++++++
 => It refers to SRS(Software Requirement Specification)
      eg: SUNMS gave JDBC API,responsible to db vendor to give the implementation
          SUNMS gave servlet API,responsible for server vendor to give the
implementation
 => It also acts like a contract b/w client and service provider
      eg: ATM Screens shows what the services the bank would offer, and bank people
would give the services what client expects.
 => Interface also refers to 100% abstract class, that is by default all the
methods present inside the interface  is
      "public and abstract".

eg#1.
interface Interf
{
      void m1();//public abstract
}
class MyImpl implements Interf
{
      void m1() //CE
      {

      }
}
```

```
eg#2.
interface Interf
{
	//public abstract
	void m1();
	void m2();
}
class MyImpl implements Interf
{
	public void m1()  //CE: class is not abstract
	{

	}
}

eg#3.
interface Left
{
	void m1();

}
interface Right
{
	void m1();
}
class MyImpl implements Left,Right
{
	//how many method to implement here ? => one method m1()
}

eg#4.
interface Left
{
	void m1();

}
interface Right extends Left
{
	void m2();
}

class MyImpl implements Right
{
	//how many method to implement here ? -> 2 methods m1() and m2()
}

eg#5.
interface Left
{
	void m1();

}
class Right
{
	public void m2()
	{

	}
```

```
}
class MyImpl implements Left extends Right //CE
{

}

eg#5.
interface Left
{
      void m1();

}
class Right
{
      public void m2()
      {

      }
}
class MyImpl  extends Right implements Left //valid
{
      public void m1(){

      }
}

Q>
interface Fish {
}
class Perch implements Fish {
}
class Walleye extends Perch {
}
class Bluegill {
}
public class Fisherman {
 public static void main(String[] args) {
      Fish f = new Walleye();
      Walleye w = new Walleye();
      Bluegill b = new Bluegill();
      if (f instanceof Perch)
            System.out.print("f-p ");//f-p
      if (w instanceof Fish)
            System.out.print("w-f ");//w-f
      if (b instanceof Fish)
            System.out.print("b-f ");
 }
}
What is the result?
A. w-f
B. f-p w-f //Answer
C. w-f b-f
D. f-p w-f b-f
E. Compilation fails.
F. An exception is thrown at runtime

QUESTION
Click the Exhibit button.
1. public class A {
```

```
2.    public String doit(int x, int y){
3.          return "a";
4.    }
5.
6.    public String doit(int... vals){
7.          return "b";
8.    }
9. }
```

Given:

```
 25. A a = new A();
 26. System.out.println(a.doit(4, 5));//a
```

What is the result?
A. Line 26 prints "a" to System.out.//Answer
B. Line 26 prints "b" to System.out.
C. An exception is thrown at line 26 at runtime.
D. Compilation of class A will fail due to an error in line 6.

Q>

```
1. public class GoTest {
2.    public static void main(String[] args) {
3.          Sente a = new Sente(); a.go();
4.          Goban b = new Goban(); b.go();
5.          Stone c = new Stone(); c.go();
6.    }
7. }
8.
9. class Sente implements Go {
10.   public void go(){
11.         System.out.println("go in Sente"); //go in Sente
12.     }
13.}
14.
15. class Goban extends Sente {
16.   public void go(){
17.         System.out.println("go in Goban");//go in Goban
18.     }
19.
20. }
21. class Stone extends Goban implements Go{  //go in Goban
22. }
23.
24.interface Go { public void go(); }
```

What is the result?
A. go in Goban go in Sente go in Sente
B. go in Sente go in Sente go in Goban
C. go in Sente go in Goban go in Goban
D. go in Goban go in Goban go in Sente
E. Compilation fails because of an error in line 17.

Answer : C

++++++++++++++++++
Exception Handling
++++++++++++++++++

=> Runtime stack mechansim
Whenever the method call happens, the stack will be used to keep the data and in a hiereachial way the stack would be formed
stack follows LIFO/FILO order to remove the stack from the JRE region.
Whenever the exception gets generated in any of the stack automatically that stack would be removed and the exception object would
be propogated to the caller,and this tracking can be seen through "Runtime Stack".

=> Default Exception Handler
The risky statements which might throw an exception can lead to the abnormal termination of the program, to avoid these in our
program we need to have an exception handling mechanism, if this is not there in our program the the exception object will be handled
by JVM specialized "Default Excpetion handler" to have smoothful termination of java program.

=> Exception hierarchy
refer paint diagram

=> control flow in try catch block
=> try with mulitple catch block
=> Need of finally block(resource releasing block)
=> difference b/w final,finally,finalize method
=> control flow in try catch and finally
=> difference b/w throw vs throws keyword
=> Exception Propogation/ducking
=> try with resource
=> Custom Exceptions(one-one interview)/Spring Rest(Global Exception handler)

eg#1.
```
try{
      System.out.println(10/0);//AE:divide by Zero
}catch(ArithmeticException e){
      System.out.println(10/0);//AE:divide by Zero
}finally{
      String s= null;
      System.out.println(s.length());//NullPointerException
}
```

A. ArithmeticException
B. RunTimeException
C. ArithmeticException and NullPointerException
D. NullPointerException//Answer
E. None of the above