**Round-1 Written test**

1. **What will be output of following program**

**public** **class** Test {

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

**int** x =9;

Boolean flag = **false**;

**if**(x<100 && x==10)

System.*out*.println(x);

System.*out*.println (x++);

**if**(x<10 && flag)

System.*out*.println (x);

}

}

1. **What will be output of this program**

**public** **class** Test {

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

**int** X=10;

String s = (X<1000)? "Small" :(X<100)? "Greater": "middle";

System.*out*.println (s);

}

}

1. **What will be output of this program**

**public** **class** Test {

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

**try** {

System.*out*.println("Divided"+6/0);

} **finally** {

System.*out*.println("Finally Block");

}

}

}

1. **What will be output of this program**

**import** java.util.ArrayList;

**public** **class** Test {

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

ArrayList al = **new** ArrayList ();

al.add ("IGATE");

al.add ("IGATEA");

al.add (1,"IGATEBB");

System.*out*.println (al);

}

}

1. **Which keyword use to prevent subclass creation**

* abstract
* finally
* super
* final

1. **Which datatype will you use to storing interest rate**

* bigDecimal
* long
* BigInt
* Float

1. **Which is the incorrect statement or throw exception, tick all if applicable**

* **int** a = Integer.*parseInt*("123.321");
* **int** b = Integer.*parseInt*("123.321f");
* **int** c = Integer.*parseInt*(123.321f);
* **int** d = Integer.*parseInt*(123.1);

1. **Display an array with all possible combinations of the strings from the given string array of four** elements [A, Ah, Aha, Aaha].

Required Output: [AA, AAh, AAha, AAaha,Aha,AhAh] distinct only.

**9-.Write the logic for reversing an array.**

**Round-2**

* Why we call thread. Start (), not thread.run (), what will happen and How it is implemented?
* Wild card, upper bound, generics
* Criteria to decide different collection
* Abstract factory pattern
* What is Enum and why it is used?

**Round-3**

* generics
* Implement your payment gateway, what are design pattern will you use, please also tell Composition, aggregation relation between classes.
* There is a 'CharacterInputStream' object in java library.It has read method which is printing all the characters read from a file.How to get only Upper case letters.Will you modify to the existing read method in it.
* What is strategic design pattern?
* Unit Testing and BDD , How to mock static object
* SOAP Web services implementation

**Round-4**

* Overloading, both are overloaded or not?

**void** add(List<Exception> a){ }

**void** add(List<String> a){}

* Different design pattern
* Executor Framework, How to identify whether all jobs has done?
* Hibernate Many to Many mapping annotation
* Hibernate Projection and interceptor
* Security in web services and different authentication technique (Basic, Digest, NTLM )

**Round-5**

* Managerial round, Project , role related question