JAVA TRAINING CENTER

Java Training Center

(No.1 in Training & placement)

Java 5(Tiger)

WorkBook

Master the Content…

W.B-1

Author

Som Prakash Rai

Topic

Jtc 1 - Jtc 4: Static Import

1) Hello.java

```
package com.Jtc.p1;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public class Hello {
public static void m1() {
System.out.println("-- m1() in com.Jtc.p1.Hello class --");
}
static void m2() {
System.out.println("-- m2() in com.Jtc.p1.Hello class --");
}
public static void m3() {
System.out.println("-- m3() in com.Jtc.p1.Hello class --");
}
public static void m4() {
System.out.println("** m4 in com.Jtc.p1.Hello class **");
}
public static void m5() {
System.out.println("== m5() in com.Jtc.p1.Hello class ==");
}
}
```

2) Abc.java

```
package com.Jtc.p2;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public class Abc {
public static void mm1() {
System.out.println("-- mm1() in com.Jtc.p2.Abc class --");
}
public static void mm2() {
System.out.println("-- mm2() in com.Jtc.p2.Abc class --");
}
static void m1Msg() {
System.out.println("-- m1Msg() in com.Jtc.p2.Abc class --");
}
public static void m5() {
System.out.println("~~ m5() in com.Jtc.p2.Abc class ~~");
}
}
```

3) Jtc1.java

```
package org.wb.test;
import static com.Jtc.p1.Hello.m1;
import static com.Jtc.p1.Hello.m4;
import static com.Jtc.p2.Abc.*;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call      :+91-9990399111
* */
public class Jtc1 {
public static void main(String[] args) {
// Hello.m1();
m1();
m4();
// m3();
mm2();
mm1();
// m1Msg()
}
static void m4() {
System.out.println("** m4 in Test class **");
}
}
```

4) Jtc2.java

```
package org.wb.test;
import static com.Jtc.p1.Hello.m1;
import static com.Jtc.p1.Hello.m4;
import com.Jtc.p1.Hello;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit        : www.jtcindia.org
*@Call        :+91-9990399111
* */
public class Jtc2 {
public static void main(String[] args) {
m1();
m4();
Hello.m4();
}
static void m4() {
System.out.println("** m4 in Test class **");
}
}
```

5) Jtc3.java

```
package org.wb.test;
import static com.Jtc.p1.Hello.*;
import static com.Jtc.p2.Abc.*;
/*
* @Author    : Som Prakash Rai
* @Join        : Java Training Center
* @visit        : www.jtcindia.org
*@Call        :+91-9990399111
* */
public class Jtc3 {
public static void main(String[] args) {
m1();
m4();
mm2();
mm1();
//m5();
}
}
```

6) Jtc4.java

```
package org.wb.test;

import static com.Jtc.p1.Hello.*;
import static com.Jtc.p2.Abc.*;
import com.Jtc.p1.Hello;
import com.Jtc.p2.Abc;
import static java.lang.System.*;
/*
```

```
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public class Jtc4 {
public static void main(String[] args) {
m1();
m4();
mm2();
mm1();
// m5();
Hello.m5();
Abc.m5();
out.println("Main Completed");
}
}
```

Jtc 5: Var-Args

```
public class Jtc5 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String[] args) {
VarArgsService service = new VarArgsService();
// service.m1();
// service.m1(12);
// service.m1(12,34,54);
service.m1(12, 43);
System.out.println();
service.m1Value(new int[] {});
service.m1Value(new int[] { 12, 23, 45 });
service.m1Value(new int[] { 12 });
service.m1Value(new int[] { 65, 78, 98, 876, 56 });
// service.m1Value(123);
System.out.println();
service.m2();
service.m2(12);
service.m2(10, 20, 30, 40);
service.m2(45, 85);
service.m2(10, 20, 30, 40, 85, 74);
service.m2(new int[] { 12, 43, 56, 78, 98 });
System.out.println();
service.add("Som", 4521.25, "Som@Jtc.org", "Som@Jtc.org","Som@gmail.com");
service.add("Som", 4521.25);
}
}

class VarArgsService {
// int... values;
void m1(int ab, int bc) {
System.out.println("-- m1(int,int) --");
}
void m1Value(int ab[]) {
System.out.println("-- m1(int[]) --");
}
void m2(int... values) {
System.out.println("-- m2(int...) --\t:" + values.length);
for (int i = 0; i < values.length; i++) {
System.out.print(values[i] + ", ");
}
System.out.println();
}
void m2(int ab) {
System.out.println("-- m2(int) --");
}
void m2(long val, double val2) {
System.out.println("-- m2(long,double) --");
}
void add(String name, double fee, String... emails) {
System.out.println("** add(String,double,String...) **");
for (String eml : emails) {
System.out.println(eml);
}
}
// void add(String name, double fee, String []emails) {}
// void m1(long...phone,String name){}
// void m2(long...phone,String... name){}
}
```

Jtc 6: Var-Args and Method Overloading

```
public class Jtc6 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
static public void main(String... args) {
VarArgsManager ref = new VarArgsManager();
ref.m1(1234);
ref.m2(1234);
ref.mm1(1234);
}
}
class VarArgsManager {
void m1(long val) {
System.out.println("--m1(long) --");
}
void m2(int... values) {
System.out.println("--- m2(int...) ---");
}
void mm1(long val) {
System.out.println("-- mm1(long) ---");
}
void mm1(int... val) {
System.out.println("-- mm1(int...) ---");
}
}
```

Jtc 7: Auto Boxing

```
public class Jtc7 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit     : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String[] args) {
Integer in = 123;// AutoBoxing
in++;
System.out.println(in + 10);
int ab = in; // AutoUnBoxing
// int ab=in.intValue();
long val = 123;
// Long ref=new Integer(123);
// Long obj=1234; // Widening and Auto Boxing
Long obj2 = 1234L; // Auto Boxing
Object obj4 = 1234;// Auto Boxing and widening
Number num = 1234; // Auto Boxing and widening
System.out.println("--------");
Integer in1 = 123;
Integer in2 = in1;
System.out.println(in1 + "\t" + in2);
System.out.println(in1 == in2);
in1++;
// int t=in1.intValue();
// t++;
// in1=new Integer(t);
System.out.println(in1 + "\t" + in2);
System.out.println(in1 == in2);
}
}
```

Jtc 8: Auto Boxing

```
public class Jtc8 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String[] args) {
Integer in1 = new Integer(123);
Integer in2 = new Integer(123);
System.out.println(in1 == in2);
Boolean b1 = new Boolean(true);
Boolean b2 = new Boolean(true);
System.out.println(b1 == b2);
System.out.println("--------------");
Boolean b3 = false;
Boolean b4 = false;
Boolean b5 = false;
System.out.println(b3 == b4);
System.out.println(b4 == b5);
System.out.println("----BYTE------");
Byte by1 = 127;
Byte by2 = 127;
System.out.println(by1 == by2);
System.out.println("-- CHARACTER --");
Character ch1 = 'A';
Character ch2 = 'A';
System.out.println(ch1 == ch2);
Character ch3 = 128;
Character ch4 = 128;
System.out.println(ch3 == ch4);
System.out.println("-- INTEGER -");
Integer in3 = 127;
Integer in4 = 127;
System.out.println(in3 == in4);
Integer in5 = 128;
Integer in6 = 128;
System.out.println(in5 == in6);
System.out.println("-- LONG -");
Long ref1 = 127L;
Long ref2 = 127L;
System.out.println(ref1 == ref2);
Long ref3 = 128L;
Long ref4 = 128L;
System.out.println(ref3 == ref4);
System.out.println("--- float ----");
Float f1 = 12.0F;
Float f2 = 12.0F;
System.out.println(f1 == f2);
System.out.println("================");
Integer in11 = 123;
Integer in12 = 122;
Integer in13 = 122;
in12++;
in13 = in13 + 1;
System.out.println(in11 + "\t" + in12 + "\t" + in13);
System.out.println(in11 == in12);
System.out.println(in11 == in13);
}
}
```

Jtc 9: Auto Boxing and Method Invocation

```
public class Jtc9 {
```

```
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call      :+91-9990399111
* */
public static void main(String[] args) {
BoxingService.m1(123);
BoxingService.m2(123);
BoxingService.m2(123);
}
}
class BoxingService {
static void m1(long val) {
System.out.println("-- m1(long)---");
}
static void m2(int... is) {
System.out.println("-- m2(int...)---");
}
static void m2(Integer in) {
System.out.println("-- m2(Integer)---");
}
}
```

Jtc 10: Auto Boxing and Method Overloading

```
public class Jtc10 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call      :+91-9990399111
* */
public static void main(String[] args) {
BoxingLoading.m1(123);
BoxingLoading.m2(123);
}
}

class BoxingLoading {
static void m1(long val) {
System.out.println("-- m1(long)---");
}
static void m1(Integer in) {
System.out.println("-- m1(Integer)---");
}
static void m2(int... is) {
System.out.println("-- m2(int...)---");
}
static void m2(Integer val) {
System.out.println("-- m2(Integer)---");
}
}
```

Jtc 11: Generics

```
public class Jtc11 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String[] args) {
Student stud = new Student();
stud.studId = "JTC-001";
stud.studId = 1245; // Auto Boxing
stud.studId = 1245L; // Auto Boxing
Student stud2 = new Student();
stud2.studId = "JTC-002";
stud2.studId = 1245; // Auto Boxing
stud2.studId = 1245L; // Auto Boxing
Student<String> st3 = new Student<String>();
st3.studId = "WB-001";
// st3.studId=4512;
Student<Long> st4 = new Student<Long>();
// st4=new Student<Integer>();
// st4.studId="WB-001";
st4.studId = 4512L;
// st4.studId=4512;

Student<?> st5 = new Student<Long>();
st5 = new Student<String>();
st5 = new Student<Integer>();

Student<? extends Number> st6 = new Student<Long>();
// st6=new Student<String>();
st6 = new Student<Integer>();
Employee emp1 = new Employee();
emp1.empId = "JTC-001";
emp1.empId = 4512;
emp1.empName = "SomPraksh";
Employee<Integer, String> emp2 = new Employee<Integer, String>();
emp2.empId = 1234;
// emp2.empId="";
System.out.println();
System.out.println();
User<Integer, Long, Address<String, Integer>> ref1 = new User<Integer, Long,
Address<String, Integer>>();
}
}

class Student<T> {
T studId;
}

class Employee<T1, T2> implements Compare<Employee<T1, T2>> {
T1 empId;
T2 empName;
// static T1 val;
public boolean compare(Employee<T1, T2> ref) {
return false;
}
}

interface Compare<T> {
public boolean compare(T ref);
}

class User<T1, T2, T3> {
T1 uid;
T2 phone;
T3 uad;
}

class Address<T1, T2> {
T1 aid;
T2 pin;
}
```

Jtc 12: Generics

```
import java.util.*;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public class Jtc12 {
public static void main(String[] args) {
System.out.println("-- Without Generics --");
List list = new ArrayList();
list.add("SomPraksh");
list.add("Manish");
list.add("Hello");
list.add("Welcome");
list.add(new Integer(12));
Iterator it = list.iterator();
while (it.hasNext()) {
Object obj = it.next();
if (obj instanceof Integer) {
Integer in = (Integer) obj;
System.out.println(in);
} else if (obj instanceof String) {
String str = (String) obj;
System.out.println(str);
}
}

System.out.println("\n-- Using Generics with List --");
List<String> list1 = new ArrayList<String>();
list1.add("SomPraksh");
list1.add("Manish");
list1.add("Hello");
list1.add("Welcome");
// list1.add(new Integer(12));

Iterator<String> it1 = list1.iterator();
while (it1.hasNext()) {
String str = it1.next();
System.out.println(str);
}
System.out.println("\n\nFrom Map Object ");
Map<Integer, String> map = new LinkedHashMap<Integer, String>();
map.put(1234, "Som");
map.put(8767, "Praksh");
map.put(5677, "Manish");
map.put(2343, "Rai");
map.put(9898, "Chandan");
Set<Map.Entry<Integer, String>> set = map.entrySet();
Iterator<Map.Entry<Integer, String>> it3 = set.iterator();
while (it3.hasNext()) {
Map.Entry<Integer, String> entry = (Map.Entry<Integer, String>) it3.next();
System.out.println(entry.getKey() + "\t" + entry.getValue());
}
}
}
```

Jtc 13 For Each Statement

```
import java.util.*;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call      :+91-9990399111
* */
public class Jtc13 {
public static void main(String[] args) {
List<String> list = new ArrayList<String>();
list.add("Welcome");
list.add("SomPraksh");
list.add(“Prakash");
list.add("Anand");
System.out.println("\n-- Using Normal For Statement (List)--");
for (int i = 0; i < list.size(); i++) {
String str = list.get(i);
System.out.println(str);
}
System.out.println("\n-- Using Normal For Statement (int[])--");
int arr[] = new int[] { 12, 34, 56, 76, 765, 33 };
for (int i = 0; i < arr.length; i++) {
int val = arr[i];
System.out.println(val);
}
System.out.println("\n-- Using Normal For Statement (int[][])--");
int dArr[][] = new int[][] { { 14, 32, 56 }, { 19, 43, 65 },
{ 98, 45, 63, 27 } };
for (int i = 0; i < dArr.length; i++) {
int[] tArr = dArr[i];
for (int j = 0; j < tArr.length; j++) {
int val = tArr[j];
System.out.print(val + "\t");
}
System.out.println();
}
System.out.println("\n*** for each statement (List) ****");
for (String s : list) {
System.out.println(s);
}
System.out.println("\n*** for each statement (int[]) ****");
for (int val : arr) {
System.out.println(val);
}
System.out.println("\n*** for each statement (int[][]) ****");
for (int tArr[] : dArr) {
for (int val : tArr) {
System.out.print(val + "\t");
}
System.out.println();
}
}
}
```

Jtc 14: Enum

```
public class Jtc14 {
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public static void main(String[] args) {
Color col = null;
```

```
// col=new Color();
Color.RED.m1Color();
Color.BLACK.m1Color();
Color.BLUE.m1Color();
Color.WHITE.m1Color();

// switch (col) {}

System.out.println("** -- ENUM -- **");
System.out.println(Days.VAL);
Days d1 = null;
// d1=new Days();

Days.SUN.m1Days();
Days.MON.m1Days();
Days.TUE.m1Days();

Object obj = Days.SUN;
Enum ref = Days.MON;

System.out.println(Days.SUN);

System.out.println("\n\n----- COURSE -------");
Course c1 = Course.JAVA;
Course c2 = Course.JDBC;
Course c3 = Course.EJB;
Course c4 = Course.JSP;

System.out.println(c1.name() + "\t" + c1.ordinal());
System.out.println(c2.name() + "\t" + c2.ordinal());
System.out.println(c3.name() + "\t" + c3.ordinal());
System.out.println(c4.name() + "\t" + c4.ordinal());

System.out.println(c1);
System.out.println(c2);
System.out.println(c3);
System.out.println(c4);
System.out.println();

switch (c1) {
case JAVA:
System.out.println("Selected course information");
System.out.println(Course.JAVA);
break;

case JDBC:
System.out.println("Selected course information");
System.out.println(Course.JDBC);
break;

case EJB:
System.out.println("Selected course information");
System.out.println(Course.EJB);
break;

case JSP:
System.out.println("Selected course information");
System.out.println(Course.JSP);
break;
}

System.out.println("\n-- Method from java.lang.Enum --");

Course courses[] = Course.values();
for (Course c : courses) {
System.out.println(c);
}
System.out.println();
String st = "JAVA";
Course cou = Course.valueOf(st);
System.out.println(cou);

}
}

class Color {
public static final Color RED = new Color("Col-001", "RED");
public static final Color BLUE = new Color("Col-002", "Blue");
public static final Color BLACK = new Color("Col-003", "Black");
public static final Color WHITE = new Color("Col-004", "White");
String colorId;
String colorName;
private Color() {
super();
}
private Color(String colorId, String colorName) {
```

```
this.colorId = colorId;
this.colorName = colorName;
}
void m1Color() {
System.out.println("\n-- m1() in Color --");
System.out.println(colorId + "\t" + colorName);
Color c1 = new Color();
}
}

interface Inter1 {
void m2();
}

enum Days implements Inter1 {
SUN, MON, TUE;
static int VAL = 90;
static {
System.out.println("-- Static Block in Days --");
}
{
System.out.println("** Instance Block in Days **");
}
Days() {
// super();
System.out.println("__ Days() Cons __");
}
public void m1Days() {
System.out.println("\n** m1Days() in Days enum**");
System.out.println("Name\t:" + name());
// Days d=new Days();
}

public void m2() {
System.out.println("-- M2 in Enum Days --");
}
}

enum E1 {}
// enum E2 extends E1{}
// enum E3 extends java.lang.Object{}
// enum E4 extends java.lang.Enum{}
```

```
enum Course {
JAVA(1, "2 Months", "SP"), JDBC(2, "7 Days", "Som"), EJB(3, "15 Days"), JSP(
4, "3 Days");

int id;
String duration;
String faculty = "SomPraksh";
static double fullCourseFee = 17000.0;

Course(int id, String duration, String faculty) {
this.id = id;
this.duration = duration;
this.faculty = faculty;
}
Course(int id, String duration) {
this.id = id;
this.duration = duration;
}
static void m1StaticMethod() {
System.out.println("-- STATIC method in ENUM--");
}
void m1CourseDetails() {
System.out.println("\n-- INSTANCE in ENUM--\t:" + this);
System.out.println("Course ID\t:" + id);
System.out.println("Duration\t:" + duration);
System.out.println("Faculty\t:" + faculty);
}
public String toString() {
return id + "\t" + name() + "\t" + duration + "\t" + faculty;
}

/*
* public boolean equals(Object obj){ return false; }
*/
}
```

Jtc 15: Formatter

```
import java.util.Calendar;
import java.util.Formatter;
/*
* @Author    : Som Prakash Rai
* @Join        : Java Training Center
* @visit        : www.jtcindia.org
*@Call        :+91-9990399111
* */
public class Jtc15 {
public static void main(String[] args) {
Formatter fmt1 = new Formatter();
String name = "SomPraksh";
int age = 23;
fmt1.format("My Name is %s and age is %d", name, age);
System.out.println(fmt1);
Formatter fmt2 = new Formatter();
fmt2.format("%c - %f - %b %n%05d - %o - %x - %X", 'C', 99.99, true,
3456, 3456, 3456, 3456);
System.out.println(fmt2);
Formatter fmt3 = new Formatter();
Calendar cal = Calendar.getInstance();
fmt3.format("%tr %n%tc %n%tl:%tM", cal, cal, cal, cal);
System.out.println(fmt3);
}
}
```

Jtc 16: Scanner

```
import java.util.Scanner;
/*
* @Author    : Som Prakash Rai
* @Join        : Java Training Center
* @visit        : www.jtcindia.org
*@Call        :+91-9990399111
* */
public class Jtc16 {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int count = 0;
double sum = 0.0;
System.out.println("Enter Numbers:");
while (sc.hasNext()) {
System.out.println("Enter Numbers Again:");
if (sc.hasNextDouble()) {
sum += sc.nextDouble();
count++;
} else {
String str = sc.next();
if (str.equals("*"))
break;
}
}
System.out.printf("Sum of given %d numbers id %f", count, sum);
}
}
```

Jtc 17: Scanner

```
import java.io.FileReader;
import java.io.FileWriter;
import java.util.Scanner;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public class Jtc17 {
public static void main(String[] args) {
Scanner sc = null;
int count = 0;
int sum = 0;
FileWriter wr = null;
try {
wr = new FileWriter("Numbers.txt");
wr.write("10 20 30 40 50 *");
wr.close();
FileReader rd = new FileReader("Numbers.txt");
System.out.println("Numbers Stored in File.");
sc = new Scanner(rd);
} catch (Exception e) {
e.printStackTrace();
}
while (sc.hasNext()) {
if (sc.hasNextDouble()) {
sum += sc.nextDouble();
count++;
} else {
```

```
String str = sc.next();
if (str.equals("*"))
break;
}
}
System.out.printf("Sum of given %d numbers id %d", count, sum);
}
}
```

Jtc 18: PriorityQueue

```
import java.util.Comparator;
import java.util.PriorityQueue;
import java.util.Scanner;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public class Jtc18 {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.println("Enter values of int type[0 - Exit]");
PriorityQueue<Integer> pq = new PriorityQueue<Integer>();
while (true) {
int val = sc.nextInt();
if (val == 0)
break;
pq.offer(val);
}
System.out.printf("Number of values stored is %d.\n", pq.size());
for (Integer in : pq) {
System.out.println(in);
}
PQSort pqs = new PQSort();
PriorityQueue<Integer> pq1 = new PriorityQueue<Integer>(8, pqs);
System.out.println("Enter values of int type[0 - Exit]");
while (true) {
int val = sc.nextInt();
if (val == 0)
break;
pq1.offer(val);
}
System.out.println("--");
System.out.println("Size\t:" + pq1.size());
System.out.println(pq1.peek());
System.out.println("Size\t:" + pq1.size());
System.out.println(pq1.poll());
System.out.println("Size\t:" + pq1.size());
System.out.println("Remaining Values in PQ1");
for (Integer in : pq1) {
System.out.println(in);
}
}
}

class PQSort implements Comparator<Integer> {
public int compare(Integer i1, Integer i2) {
return i1 - i2;
}
}
```

Jtc 19: Annotation (Override & Deprecated)

```
import java.util.Date;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */

public class Jtc19 {
public static void main(String[] args) {
Student st = new Student();
System.out.println(st);
st.m2Info();
st.m1StudentInfo();
Date dt = new Date();
System.out.println(dt.getDate());
System.out.println(dt.getMonth());
System.out.println(dt.getYear());
}
}

class Student {
@Override
public boolean equals(Object obj) {
return super.equals(obj);
}
@Deprecated
void m1StudentInfo() {
}
void m2Info() {
System.out.println("-- M2Info --");
}
public int hashcode() {
return 10;
}
/*
@Override
public String tostring() {
return "Student Obj";
}
*/
}
@Deprecated
class Hello{}

@Deprecated
interface Inter1{}
```

Jtc 20: Annotation (SuppressWarnings)

```
import java.util.Date;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call      :+91-9990399111
* */
public class Jtc20 {
@SuppressWarnings({ "deprecation", "static-access", "unused" })
public static void main(String[] args) {
```

```
int ab = 10;
Date dt = new Date();
System.out.println(dt.getDate());
System.out.println(dt.getDate());
Employee emp = null;
emp.m2();
}
}

class Employee {
void m1() {
@SuppressWarnings("unused")
int xy = 10;
int mn = 90;
}
static void m2() {
System.out.println("** M2 Static Method **");
}
}
```

Jtc 21: Annotation (Custom Annotation)

```
import java.lang.annotation.*;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
public class Jtc21 {
public static void main(String[] args) {
@Author(id = 101, name = "SomPraksh", phone = 8585745L)
Book b1 = new Book("Core Java");

@Author(name = "Vikas", phone = 9673563844L)
Book b2 = new Book("Hibernate");
System.out.println("---- APT ------");
boolean pre = b2.getClass().isAnnotationPresent(BBAnno.class);
if (pre) {
BBAnno ref = (BBAnno) b2.getClass().getAnnotation(BBAnno.class);
System.out.println("Value\t:" + ref.value());
}
Annotation ans[] = b2.getClass().getAnnotations();
for (Annotation an : ans) {
System.out.println(an);
}
System.out.println();
JtcServlet serv = RegisterServlet.class.getAnnotation(JtcServlet.class);
System.out.println(serv.url());
}
}

@Target(ElementType.TYPE)
@interface JtcAnno { }

@Retention(RetentionPolicy.RUNTIME)
@interface BBAnno {
String value();
}

@Retention(RetentionPolicy.RUNTIME)
@interface JtcServlet {
String url();
}

@Target(ElementType.LOCAL_VARIABLE)
@interface Author {
int id() default 99;
String name();
long phone();
}

@JtcAnno
@BBAnno("1234")
class Book {
// @JtcAnno
String name;
Book(String name) {
this.name = name;
}
}

@BBAnno(value = "1234")
class BBStudent { }
```

```
/*
* @JtcServlet("/login.Jtc") class LoginServlet{}
*/

@JtcServlet(url = "/regster.Jtc")
class RegisterServlet { }
```

Jtc 22: Annotation (Custom Annotation)

```
import java.lang.annotation.*;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call       :+91-9990399111
* */
class Jtc22 {
public static void main(String[] args) throws CloneNotSupportedException {
Emp emp = new Emp(99, "SomPraksh");
System.out.println(emp);
Emp emp2 = emp.getClonedObject();
System.out.println(emp2);
System.out.println(emp == emp2);
}
}

@JtcCloneable
class Emp {
int eid;
String ename;

Emp(int eid, String ename) {
this.eid = eid;
this.ename = ename;
}

@Override
public String toString() {
return eid + "\t" + ename;
}
public Emp getClonedObject() throws CloneNotSupportedException {
boolean b1 = this.getClass().isAnnotationPresent(JtcCloneable.class);
if (b1) {
return new Emp(this.eid, this.ename);
}
throw new CloneNotSupportedException("Emp class not using JtcCloneable");
}
}

@Retention(RetentionPolicy.RUNTIME)
@interface JtcCloneable { }
```

Jtc 23: Annotation (Custom Annotation)

```
import java.lang.annotation.*;
import java.lang.reflect.*;
/*
* @Author    : Som Prakash Rai
* @Join      : Java Training Center
* @visit      : www.jtcindia.org
*@Call      :+91-9990399111
* */
public class Jtc23 {
public static void main(String[] args) throws Exception {

JtcEmployee emp = new JtcEmployee();
emp.empId = 99;
emp.empName = "SomPraksh";
emp.empPhone = 6526668;
JtcJdbcTemplate res = new JtcJdbcTemplate();
res.save(emp);

JtcStudent stud = new JtcStudent();
stud.studId = 3131;
stud.studName = "Manish";
stud.phone = 9590712983L;
stud.studFee = 27000.0F;
res.save(stud);
}
}
```

```
@Table(name = "studTable")
class JtcStudent {
@Column(name = "sid")
int studId;
@Column(name = "sfee")
float studFee;
long phone;
@Column(name = "sname")
String studName;
}

@Table(name = "empTable")
class JtcEmployee {
@Column(name = "eid")
int empId;
@Column(name = "eage")
int age;
@Column(name = "ename")
String empName;
@Column(name = "ephone")
long empPhone;
}

// IMPLEMENTED BY JTC VENDOR

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE)
@interface Table {
String name();
}

@Target(ElementType.FIELD)
@Retention(RetentionPolicy.RUNTIME)
@interface Column {
String name();
}

class JtcJdbcTemplate {
public void save(Object obj) throws Exception {
boolean tabPresent = obj.getClass().isAnnotationPresent(Table.class);
if (tabPresent) {
Table tab = obj.getClass().getAnnotation(Table.class);
String tableName = tab.name();
String qry = "insert into " + tableName;

StringBuffer cols = new StringBuffer(" (");
StringBuffer values = new StringBuffer(" values (");

Field fs[] = obj.getClass().getDeclaredFields();
for (int i = 0; i < fs.length; i++) {
Field f1 = fs[i];
boolean colPresent = f1.isAnnotationPresent(Column.class);
if (colPresent) {
String colName = f1.getAnnotation(Column.class).name();
cols.append(colName + ",");
String type = f1.getType().getSimpleName();
if (type.equals("int")) {
int val = f1.getInt(obj);
values.append(val);
values.append(",");
} else if (type.equals("String")) {
String val = f1.get(obj).toString();
values.append("'");
values.append(val);
values.append("',");
} else if (type.equals("long")) {
long val = f1.getLong(obj);
values.append(val);
values.append(",");
} else if (type.equals("float")) {
float val = f1.getFloat(obj);
values.append(val);
values.append(",");
}
}
}
qry = qry + cols.substring(0, cols.length() - 1) + ")"+ values.substring(0, values.length() - 1)
+ ")";
System.out.println(qry);
// Code to Store in DB
} else {
String cName = obj.getClass().getName();
throw new RuntimeException(cName + " is not annotated with @Table");
}
}}
```