STATIC VERSUS NON STATIC

Non Static:

1. Non static members belongs to object and when ever an object is created non static member will get

loaded into the object

2. Number of objects created will equal to number of copies of non static member

3. Without creating objects non static members cannot be used

static:

1. Every class by default has dedicated commom memory given. When a member is made static it will

automatically get loaded into the common memory of the class and hence it belongs to class

2. static members are loaded into the common memory only once

3. Static variables value can be changed

Example 1:

```
public class A {

static int i = 500;

static int j = 1000;
int k = 100;

public static void main(String args[]) {

A a1 = new A();

System.out.println(a1.k);

System.out.println(A.i);

System.out.println(A.j);

}

}
```

Output:

```
100

500

1000
```

Example 2:

```
public class A {
```

```
int i = 100;

public static void main(String args[]) {

A a1 = new A();

A a2 = new A();

System.out.println(a1.i);

System.out.println(a2.i);



}

}
```

Output:

```
100
100
```

Example 3:

```
public class A {



int i = 100;
public static void main(String args[]) {

A a1 = new A();

 A a2 = new A();

System.out.println(a1);

System.out.println(a2);



}

}
```

Output:

```
A@7960847b

A@6a6824be
```

Example 4:

```
public class A {

static int i = 100;
public static void main(String args[]) {

System.out.println(A.i);//100

A.i = 500; //100 is replaced with 500 in i

System.out.println(A.i);

}

}
```

Output:

```
100
500
```

Example 5:

```
public class A {

int i = 100; //non static

public static void main(String args[]) {

 A a1 = new A();
System.out.println(a1.i);

a1.i = 500; // 100 is being replaced with 500 in variable i

System.out.println(a1.i);



}


}
```

Output:

```
100

500
```

Lets understand static methods and non-static methods in java:

____

Example 1:

```
public class A {



public static void main(String args[]) {



A a1 = new A();
a1.test();



}



public void test(){ //non static
  method belongs to object
System.out.println(1000);

}

}
```

Output:

```
1000
```

```
public class A {



public static void main(String args[]) {



A.test();


}
public static void test(){ //static

  method belongs to class

System.out.println(1000);


}


}
```

Output:

```
1000
```

Types of variables in Java ?

1. local variables

a. Local variables are created inside a method

b. Local variables can be used only within the created method, outside that method variable is not

accessible

c. To access local variable you need not create object nor use class name, these variables are

accessed directly with its name

d. Without initalizing local variables if used then you will get an error

Example 1:

```
public class A {
```

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

System.out.println(i);
  //Error

A a1 = new A();

a1.test();

}



public  void test(){

int i = 10;

System.out.println(i);



}

}
```

Out put: Error

Example 2:

```
public class A {



public static void main(String args[]) {

int i = 10;

System.out.println(i);



A a1 = new A();

a1.test();

}



public  void test(){

System.out.println(i);
   //Error

}

}
```

Output: Error

Example 3:

```
public class A {



public static void main(String args[]) {

int i = 10;

System.out.println(i);



}



}
```

Output:

```
10
```

Example 4:

```
public class A {
```

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

A a1 = new A();

a1.test();


}


public void test(){

int i = 500;

System.out.println(i);

}


}
```

Output:

```
500
```

Example 5:

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

int i;

System.out.println(i);

}

}
```

Output:

Error

2. static variables

3. non-static variables

4. reference variables in depth