Java Generics

→ Introduced from JDK 5.0 onwards
→ They help us to ~~deal~~ deal with the compiler time type-safety.
→ With the help of the Generics, we can ~~write~~ write a single method & call it with diff. arguments types (Integer, strings).

Advantages of Generics:

Bugs can be detected at compile time:

→ While developing any application or program, it is always better to catch the bug / problem at the compile time instead of runtime so that we can provide a smooth experience to the user.

E.g.

```
import java.util.ArrayList;

public class CWH {
    p s v m ( ) {
        // Without Generics

        ArrayList array = new ArrayList();
        array.add(10);
        array.add("Edward");
        array.add(20.4);
        sout(array);
    }
}
```

o/p

~~10~~ [10, Adwait, 20.4]

~~Adwait~~

Here, we can notice that we can store any type of object in a collection without Generics. But, with Generics we can store only one type of object.

E.g.

```
import java.util.*;

public class CWH{
    p s v m(){
    //With Generics

ArrayList<Integer> Array = new
                           ArrayList();

 Array.add(10);
 Array.add("Adwait");
 Array.add("20.4");
 sout(Array);
    }
}
```

o/p:

Java: incomp. types java.lang.String cannot be conv. to j.l.Int.

" ——— " double " ——— "

Hence the code process incompatible type error.

Type-casting not required:

→ Let's suppose you created an arrayList (without Generics), & you want to store the value at index 0 into a Integer variable x, but you can't do it in java ∵ arrayList returns an object, but we're storing the value in a Integer variable. In such cases, we need to typecast the object in the desired data type. But with Generics we don't need to type cast.

E.g.

```
import java.util.*;
    psvm ( ){

  ArrayList arr = new ArrayList();
    arr.add(10);
    arr.add("Adwait");
  int x = (int) arr.get(0);
    int x = arr.get(0);
      sout(x);
    }
}
```

o/p:

```
java: incompatible types: java.lang.Object
        can't be converted to int.
```

o/p

10

With the help of Generics

E.g. Code

```
import java.util.*;

public class CWH{
    psvm(){

ArrayList<Integer> arr = new ArrayList();
    arr.add(10);
    arr.add(20);

int x = arr.get(0);
  sout(x);
 }
}
```

o/p:
10