NOTES TAKING - 09-04-2023

wrapper class	(primitive type)
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

Return statement :

end of the method. it return value

Void :

does not return any ~~type~~ thing

Parameterized Method :

~~Access~~ Access single variable and its values by multiple objects.

parameter (or) argument

Access specifier :

default :

within the package.

float, double,

Day-4-09-04-23

wrapper class:- (class of datatypes)

the wrapper class in Java provides the mechanism to convert primitive into object and object into primitive.

Primitive Type	wrapper class.
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double.

Return Statement:-

Returning a value from method.

In java every method is declared with a return type, such as int, float, double, string. There return types required a return statement at the end of method.

The return keyword is used for returning the resulted value.

If we try to return a value from void method the compiler shows an error

Example program:

```
package Org.datatypes;
public class TypesofVariables
{  int a=60;
   int b = 50;
  public int addition()
    { int c=a+b;
      System.out.println(c);
      return c;
    }
  public static void main
      (String [] args) {
  Types of Variable tv =
      new  Types of Variables();
  int  result = tv.addition();
  System.out.println(result
                  +90);
 }
}
```

PARAMETERS AND ARGUMENTS:-

Information can be passed to methods as parameter.

you can add as many parameters as you want, just separate them with a comma.

when a parameter is passed to the method, It is called an argument.

<u>SYNTAX:-</u>

```
public void methodName(arg1, arg2)
{ }
```

call parameterized method :

```
objectreferancename . methodname (parameter1,
    Parameter2)
```

Example Program

```
Package org . parameterized;
package org. variable . types;
public class ExampleParameterized
{
     String name;
     int id;
     String Qualification;

     Public void studentDetails()
     {
           System.out. println
                          (name);
```

Scanned with CamScanner

```
System.out.println (id);
System.out.Println (Qualification);
 }
public Static void main (
String [] args)
{  ExampleParameterized ep=
new   ExampleParameterized();
 ep.name = "Aravind";
 ep.id = 11;
 ep.Qualification = "BE";
 ep.studentDetails();

 ExampleParameterizd ep2 =
  new  ExampleParameterized();
 ep.name = "Rajitha";
 ep.id = 44;
 ep.Qualification = "BTech"
 ep. studentDetails();

}
 }
Package.ins.firstday;

public class StudentDetails {

  public void StudDetails
  (String name, int id, int marks)
   { System.out.println
                 (name)
```

```
System.out.println(id);
System.out.println(marks);
}
public static void main(String[] args)
{   StudentDetails s = new StudentDetails();
    s.studentDetails("Dilip", 11, 234);
    s.studentDetails("Khan", 22, 123);
    StudentDetails S = new StudentDetails();
    S2.studeDetails("shobana", 44, 566);

}
}
```

```
Package org.parameterized;
   public class Student
   { Package org.variable.types;
public class ExampleParameterized
  { int a;
    public void studentDetails
  (String name, int id, String
                    Qualification){
    System.out.println(name)
   System.out.Println(id);
   System.out.println(Qualification);

 }
public static void main
(String [] args)
{ ExampleParameterized ep =
new ExampleParametrized();
ep.studentDetails ("Aravind",
                  "", "B.E");
ep.studentDetails ("Faizal", 22,
                  "BTECH");
ExampleParameterized ep1 = new
        ExampleParameterized();
ep1. StudentDetails ("Priyanka",
        33, "Rajitha");
}
}
```

Access specifier.

In Java, the access specifier used for "restricting the scope" of a class and its data members, member function and constructor

(i) Private
(ii) protected
(iii) public
(iv) default

PRIVATE:-

The private access ~~identifier~~ modifier is accessible only within the class.

PROTECTED:-

We can access the protected data members and member function of a class within the same package or subclasses in different packages

PUBLIC:-

We can access the methods and variables of a class from anywhere in the program, within is declared as public In simple words, no Restriction is allowed on scope of public data member.

DEFAULT:-

If you don't use any specifier it is treated as default by default.

The default modifier is accessible only within package. It cannot be accessed from outside the package.

It provide more accessability than Private But, it is more restrictie than protected and public

Important Points

Access Specifiers define the Visibility of the class.

If no keyword than that is default access modifier.

Four modifiers in Java include public, private, protected and default

Private and protected keywords cannot be used for classes and interfaces

Access Modifier	Within class	within package	outside package by subclass only	Outside Package
Private	Y	N	N	N
Defaut	Y	Y	N	N