FIRST JAVA PROGRAM

Structure of Java File

"Source code that we write will be saved using extension .java"

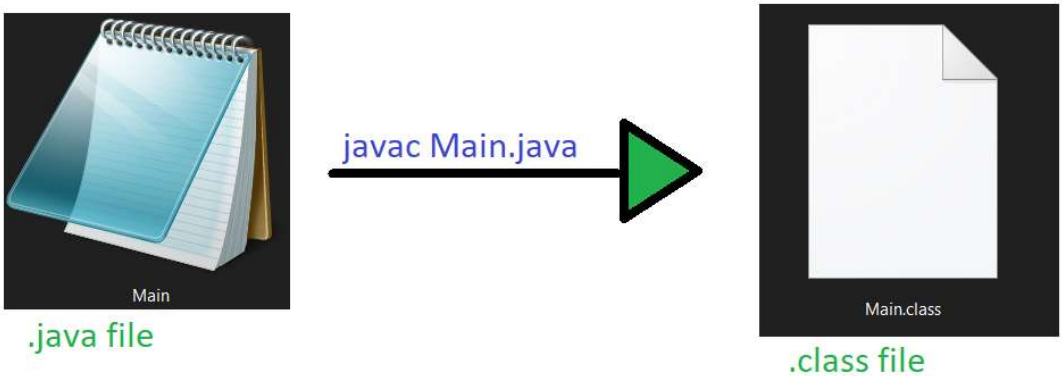
- Every thing written in .java file must be in classes or we can say that every file having .java extension is a class
- A class with same name as file name must be present in .java file.

First alphabet of class name can be in upper case. It is the naming convention of class name. however, it is not compulsory to do so.

- Class which is having same name as file must be public class
- A main function/method must be present in this public class, main is a function from where the program starts.

Converting .java to .class

- Using javac compiler we can convert .java file to .class
  Command to convert .java to .class
  Javac and .java file name
  Let the name of .java file is Main, so the command to convert .java to .class is
  Javac Main.java
- Above command create a . class file (Main.class) which contains Bytecode.



Running the program

By using java and name of file we can run the program.

- Command > **java Main**

Hello world program

```
public class Main{
    public static void main(String [] args){
        System.out.println("Hello World");
    }
}
```

1. **public (in first line):-** public is an access modifier which allows to access the class from anywhere.
2. **class :-** It is a name group of properties and functions
3. **Main :-** It is just the name of class as same as the name of file.
4. **public (in second line) :-** It is used to allow the program to use main function from anywhere.
5. **static :-** It is a keyword which helps the main method to run without using objects.
6. **void :-** It is a keyword used when we do not want to return anything from a method/function
7. **main :-**It is the name of method.
8. **String [] args :-** It is a command line argument of string type array.
9. **System:-** It is a final class defined in java.lang package.
10. **out :-** It is a variable of PrintStream type which is public and static member field of the System class.
11. **println :-** It is a method of PrintStream class, It prints the argumants passed to it and adds a new line. **print** can aalso be used here but it prints only arguments passed to it. It do not adds a new line.

What is package ?

→ It is just a folder in which java files lies.

→ It is used to provide some rules and stuff to our programs.

Primitive data types

- Primitives data types are those data types which is not breakable.

Ex:-
String is not a primitive data type so we can break this data type into char

i.e., String “Kunal” can be divided into

‘K’ ‘u’ ‘n’ ‘a’ ‘l’

But primitives data type are not breakable.

We cannot break a char ,int etc.

- List of primitive data types in java are :-

Data types	Description	Example
int	int is used to store numeric digits	int i = 26;
char	char is used to store character	char c = ‘A’;
float	flot is used to store floating point numbers	float f = 98.67f;
double	double is used to store larger decimal numbers	double d = 45676.58975 ;
long	long is used to store numeric digits which is not able to stored in int	long l = 15876954832558315l;
boolean	It only stores store t values i.e., true or false.	boolean b = false;

In float and long we have used f and l, it denotes that the number in the variable is float or long type, if we do not use this java consider float value as double and long value as int.

- Literals :- It is a synthetic representation of boolean, character, string, and numeric data.

  Ex:- int a = 10;

  Here 10 is called literal.
- Identifiers:- name of variable, methods, class, packages, etc. are known as identifiers.

  Ex:- int a = 10;

  Here a is Identifier.

Comments in Java

Comments are something which is written in source code but ignored by compiler.

- Two types of Comment
  1. Single line comment :- used to comment down a single line (// is used for it.)
  2. Multi line comment :- used to comment down multiple lines (/* */ is used for it)

Inputs in Java

We have Scanner class available in java.util package to take input

To use this class we have to

1. Import java.util package in our file.
2. Create object of the scanner class
3. Use that object to take input from the keyboard.

Syntax :-

```
import java.util.Scanner;
public class Main{
    public static void main(String [] args){
        Scanner input = new Scanner(System.in);

    }
}
```

1. **Scanner :-** It is a class required to take input, it is present in java.util package.
2. **input :-** It is an object that we are creating to take input.
3. **new :-** It is a keyword used to create an object in java.
4. **System.in :- System** is a class and **in** is a variable that denotes we are taking input from standard input stream (i.e. Keyboard).

int Input :- nextInt() is a function used to take input of int.

Syntax:-

```
Scanner input = new Scanner(System.in);
        int rollno = input.nextInt();
```

float Input :- nextFnt() is a function used to take input of int.

Syntax:-

```
Scanner input = new Scanner(System.in);
        float marks = input.nextFloat();
```

String Input :- Two ways to take string input

1. Using next() Method :- It will take one word input till a space occurs

   Syntax:-

```
Scanner input = new Scanner(System.in);
        String s1 = input.next();
```

   Input :- Hey kunal

   Output :- Hey

2. Using nextLine() Method :- It will take all string input including space.

   Syntax:-

```
Scanner input = new Scanner(System.in);
        String s2 = input.nextLine();
```

Sum of two numbers

```java
import java.util.Scanner;

public class Sum {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter first number");
        int num1 = input.nextInt();
        System.out.print("Enter second number");
        int num2 = input.nextInt();
        int sum = num1+num2;
        System.out.println("Sum = "+sum);

    }
}
```

Output

```
Enter first number70
Enter second number80
Sum = 150
```

Type conversion kisko kehta hai
Type casting mein aap kisko cast karte ho.

Type conversion

When one type of data is assigned to another type of variable an automatic type conversion will take place under some condition

Conditions :-

1. Two types should be compatible.
2. Destination type should be greater then the source type.

Type Casting

When we convert one type of data to another type is known as type casting

Ex:- int num = (int) (67.564f)

Automatic type promotion in expressions.

While evaluating expressions the intermediate value may exceed the range of operands and hence the expression value will be promoted.

There are some condition for type promotion:-

1. Java automatically promotes each byte, short or char operand to int when evaluating an expression.
2. If one operand is a long , float or double the whole expression is promoted to long , float or double respectively.

Ex:-

```
byte a = 40;
byte b = 50;
byte c = 100;
int  d = (a*b)/c;
System.out.println(d);
```

Here when a*b occurred it became 2000 which is out of the range of byte so here byte is automatically promoted to int type.

Example for thorough review concept.

```
public class TypePromotion {
    public static void main(String[] args) {
        byte b = 42;
        char c = 'a';
        short s = 1024;
        int i = 50000;
        float f = 5.67f;
        double d = 0.1234;
        double result = (f*b)+(i/c)-(d*s);
        System.out.println((f*b)+" "+(i/c)+" "+" "+(d*s));
        System.out.println(result);

    }
}
```

Output

```
238.14 515   126.3616
626.7784146484375
```

Prime number program.

```java
import java.util.Scanner;
public class Prime {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Please enter a number");
        int n = in.nextInt();
        if(n<=1){
            System.out.println("Nither prime nor composite");
            return;
        }
        int c=2;
        if(n==4){
            System.out.println("Not Prime");
        }
        else{
            while(c*c<n){
                if (n%c==0){
                    System.out.println("Not Prime");
                    return;
                }
                c=c+1;
            }
            if(c*c>n){
                System.out.println("Prime");
            }
        }
    }
}
```

Output :-

1.
```
Please enter a number
17
Prime
```

2.
```
Please enter a number
1
Nither prime nor composite
```

3.
```
Please enter a number
6
Not Prime
```

Example of if statement.

Statement inside if statement only executes when condition given in if is true.

```
public class ifstatement {
    public static void main(String[] args) {
        int a = 10;
        if (a == 10){
            System.out.println("Hello");
        }
    }
}
```

output

```
Hello
```

Example of while loop.

Statement in while loop run till condition in while loop become false

```
public class whileloop {
     public static void main(String[] args) {
            int count = 1;
           while (count != 5) {
            System.out.println("count");
            count++;
         }
     }
 }
```

Output

```
count
count
count
count
```

Example of for loop.

```
public class forloop {
    public static void main(String[] args) {
        for (int count=1;count!=5;count++){
            System.out.println(count);
        }
    }
}
```

Output

Celsius to Fahrenheit program.

```
import java.util.Scanner;

public class CelsiusToFahrenheit {
    public static void main(String[] args) {
        Scanner in = new Scanner (System.in);
        float tempC = in.nextFloat();
        float tempF = (tempC*9/5)+32;
        System.out.println(tempF);
    }
}
```

Output