L1. First Java Program

1. First Java Program

Structure of a Java Program

i. Main classes in java should have the same name as the file.

- It is a good practice to have the name of the main class start with a capital letter. This is the correct convention.
  Eg. `class Main{...}`
- `public class Main{...}` in the line given here "public" is an access modifier and "Main" is the class name.
- The access modifier public allows the class to be accessed anywhere in the program.

ii. Now inside the file we should create the main function,

- A **function** is a block of code that can be run multiple times on calling the function name, this allows code reusability.
  Eg. `public static void main(String[] args){...}
- The above line of code is the main function of the program, this contains the main code for the program.

iii. Output Statements

- An output statement is a line of code that is used to display information onto the console.
  Eg. `System.out.println("Hello World!")`

What is 'static'?

- Static is a **keyword** that is used to say that a method or variable belongs to a **class** and not an object of the class.
- It is used in the main function public static void main(String[] args) this is so that JVM can call it without having to create and object of the class. This is not possible because the program starts from the main class and no objects can exists yet when it is first called.

What is void?

- "void" is a **return type** of a function.
  Eg. When we want the sum of two numbers using a function we can use the int return type.
- "void" is a return type that does not return anything (NULL).

Array and Arguements

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

- "String[] args" is an array for arguments to be passed from the terminal to the main function before the execution of the program.
- "args" is the name of the String and is short for arguments.
- Below is the right method to pass arguments to the Main function.

  “Pasted image 20250625194725.png” could not be found.

Code for the program to print the argument in the 0th index:
`public class Main {` public static void main(String[] args) {
System.out.println(args[0]); } }`

O/P: Gabriel

Changing the Location of Byte Code

- Byte Code is stored in a .class file.

  Commands:
  1. To store in a current file:
     "javac -d.Demo.java.c"

2. To store in the previous directory:
   "javac -d.Demo.java"

Location of 'javac' and 'java'

Where is javac and java located?
~It can be located using the commands in the terminals:
`where javac or java`
O/P: Gives the location of the file.

What are Packages?

Packages are folders that are used for organizing java code.
__Purpose:

- Organizing Code
- Access Control (Visible/Not Visible outside classes)
- Reusability - Groups reusable classes or libraries.
- Avoids naming conflicts

Output Statements

`System.out.println("...")`
The above line is an output statement where,
"System" - is a class
"out" - is a type
"println()" - is a method

System is a class which has the variable called out, of type the print stream.
Out has a method called println().

Inputs in java

To pass an inputs in java we need to import the **Scanner** class from the **utilities** package in java.
`import.util.Scanner;`

Creating a scanner object to read input,
`Scanner input = new Scanner(System.in)`

"input" - object of scanner class
new - keyword for creating object
System.in - Input Stream

Input can be take from the keyboard and also from a file.

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

Primitive Data types in java

Primitive data types are the base data types in java which cannot be broken down into any other smaller data type.
Eg. Integers, floats and characters

An example of a non primitive data type is a String whose index is from 0 to N-1 and each index stores a character which can be accessed individually using its respective index.

Primitive Data types space
Integer: int rolno. = 64; -> 4 bytes
Character: char letter = 'r'; -> 2 bytes
Float: float salary = 9000.99f; -> 4 bytes
Double: double Ldecimal = 4567845678.45678; -> 8 bytes

long: Linteger = 234567890098765432L; -> 8 bytes
Boolean: boolean check = false;

Comments in Java

Lines in a program that are ignored by the compiler on execution of the program are called comments.
Eg. `//This is a comment.`

Multiline Comments:

```
/* BLAH
     BLAH
        BLAH
*/
```

Integer Input in java

```
public class Main{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("Please Enter a integer input: ");
        int rollno = sc.nextInt();
        System.out.println("Your roll no is: " + rollno);

    }
}
```

int a = 10;

To accept data types such as:
Strings -> String name = sc.nextLine();
Float -> float gpa = sc.nextFloat();

Type Casting and Conversion

Type Conversion

When a certain type of data is assigned to another type of variable an automatic type conversion takes place if the following conditions are met.
Eg. Inputting A integer value into a float variable and on printing the variable we get a float value.

For type conversion there are **two conditions**:

1. The two types should be compatible.

2. The LHS variable should be greater than the RHS value
   Eg. floats can have whole numbers with decimals but integers can not.

Type Casting

Compressing a bigger number to a smaller type explicitly.
Eg. Converting float to integer.
``int num = (int)(67.46f);

The above line of code turns the float data type into an integer data type.
Note: This can be done for byte as well.

Automatic Type Promotion in Expressions

Assume that the byte has a range of 256.

```
// Case 1:
int a = 257;
byte b = a; // A is an integer

Case 2:
byte b = (byte)(a);
\\ On printing b we get 1 after casting.
```

What it does is take the value that you have given and then find its remainder using the max possible value!
Eg. Giver_Val % Max_val
257 % 256

Case 3:

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

You may ask how is this possible
40 x 50 = 2000
byte x byte = byte

Byte can not hold 2000
But, this is possible because the byte evaluations are done in integer since "int d"

Case 4:

```
byte b = 50;
b = b*2;
```

- The above case is actually an error, this is because an evaluation has to be done in integer format.
- The result of the evaluation can not be assigned back without **explicit casting**.

Case 5:

```
int number = 'A';
System.out.println(number);
```

- The above code shows that a character 'A' is assigned to the another number because of type conversion.
- On printing number we get 65 since, this is the ASCII value of 'A'.

Note: Java follows Unicode thus all languages such as Hindi can be printed in java.

Rules for type promotion in java

All byte. short, char values are promoted to integers.
If any operand is of a long type then the whole operation is promoted to long, the same goes for floats and doubles. Basically the result of smaller and larger values is given as larger.

Note: When we don't know how many times the loop is going to run then we use a while loop, If we do know how many times the loop will run we use a for loop.

Formula for Temperature Conversion: Fahrenheit = ((Celcius x 9/5)+32)