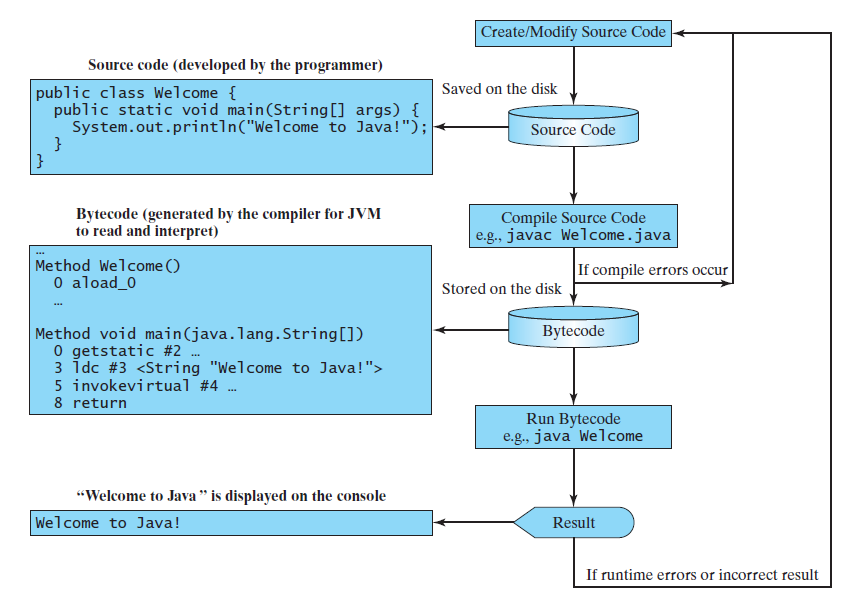
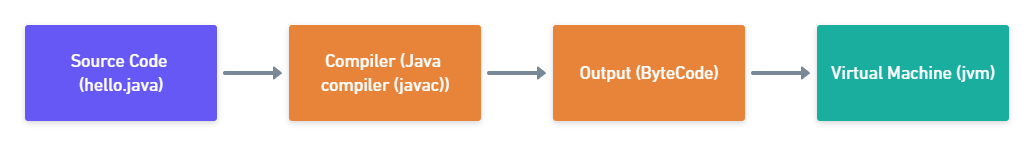
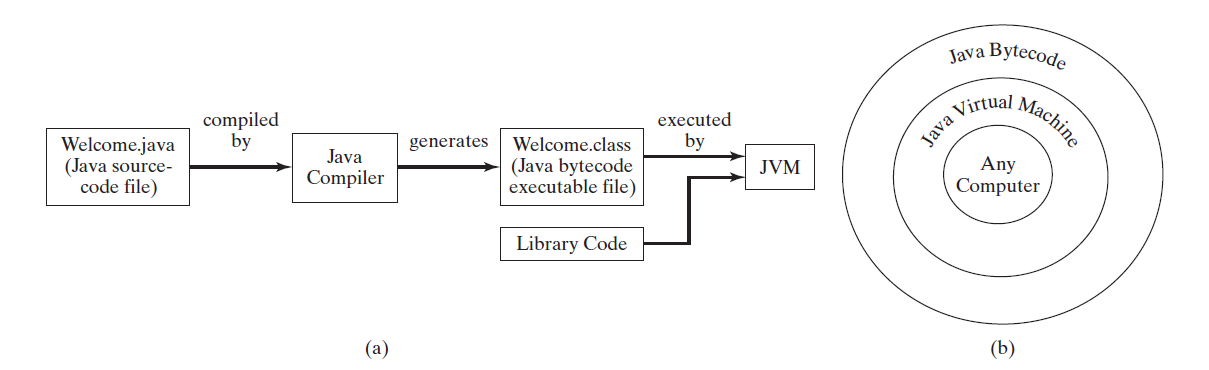
**Java**

**1 Introduction**

Java is a pure object-oriented language given by sun microsystems in 1991 which Oracle purchased in 2010.







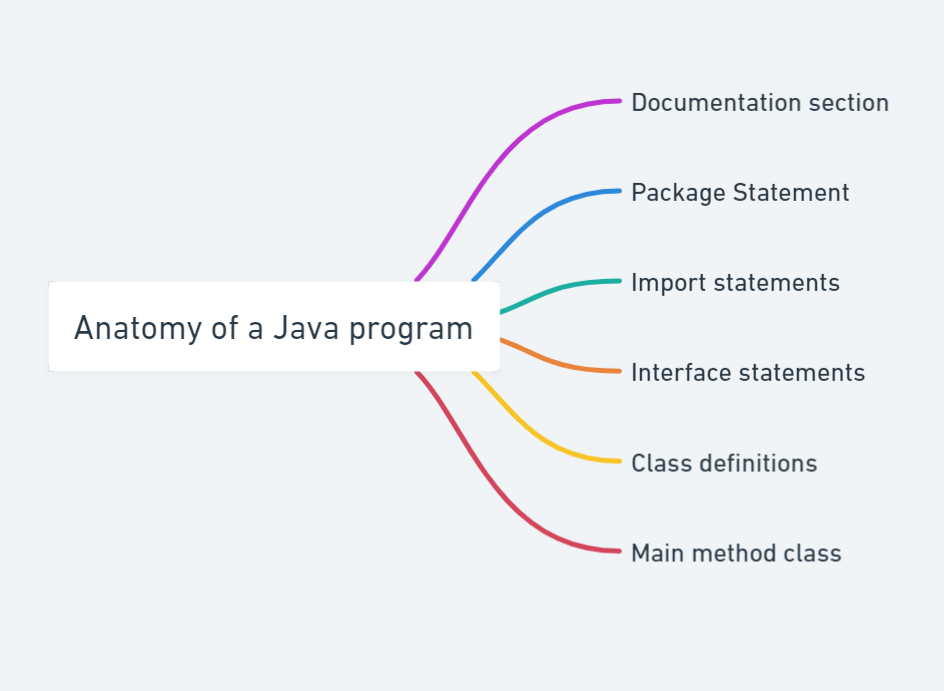
javac hello.java //compiles

java Hello //executes

* 1. **Basic structure of the program**

package com.company; //used to group related classes  
  
public class Main { // entry point to application  
  
 public static void main(String[] args) {  
 // write your code here  
 System.out.println("Hello World");  
 }  
}

1. Functions (camelCaseConvention for naming)
2. Classes (PascalConvention for naming)



* 1. **Variables and Datatypes**

1. Primitive Data types (Intrinsic)
2. Non – Primitive Data Types (Derived)

For output,

System.out.println("Hello World");  
System.out.printf("%d\t%d\t%d",i,i\*i,i\*i\*i);

Note: char takes 2 bytes (Unicode)

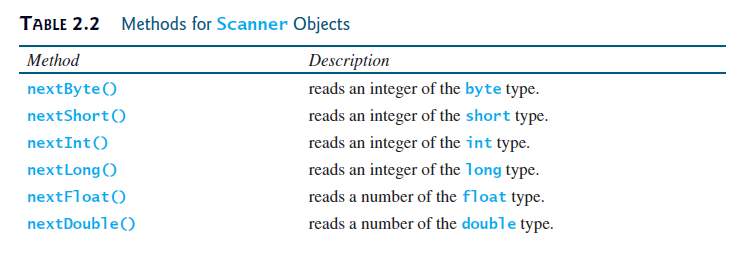
[Flowchart for datatypes](https://whimsical.com/datatypes-8jurkPiVZH1hCiFric1NP5)

**Literals:** a constant value which can be assigned to a variable is called literal.

**Keywords:** Words which are reserved by the java compiler

**Identifier:** Identifiers are the names that identify the elements such as classes, methods, and variables in a program.

**Getting user input**

****

Scanner class is used for this purpose

package com.company;  
  
import java.util.Scanner;  
  
public class Tut5\_TakingInput {  
 public static void main(String[] args) {  
 Scanner S = new Scanner(System.in);  
 System.out.println("Enter integer");  
 int a=S.nextInt();  
 System.out.println("Enter float");  
 float x=S.nextFloat();  
 float sum=a+x;  
 System.out.println("Sum is:"+sum);  
 boolean b1=S.hasNextInt();  
 System.out.println("Is next input integer?"+b1);  
  
 S=new Scanner(System.in);  
 System.out.println("Input a string:");  
 String str1=S.next();  
 System.out.println("String is:"+str1);  
  
 S=new Scanner(System.in);  
 System.out.println("Input a string:");  
 String str2=S.nextLine();  
 System.out.println("String is:"+str2);  
 }  
}

Note: Before taking string input always flush buffer using

sc = new Scanner(System.in);

**Q1: cbseCalculator**

package com.company;  
  
import java.util.Scanner;  
  
public class Tut6\_calc {  
 public static void main(String[] args) {  
 int eng,maths,sci,sst,sans;  
 Scanner sc = new Scanner(System.in);  
 System.out.println("Enter the marks of Eng:");  
 eng = sc.nextInt();  
 System.out.println("Enter the marks of maths:");  
 maths = sc.nextInt();  
 System.out.println("Enter the marks of sci:");  
 sci = sc.nextInt();  
 System.out.println("Enter the marks of sst:");  
 sst = sc.nextInt();  
 System.out.println("Enter the marks of sans:");  
 sans = sc.nextInt();  
 float res = (eng+maths+sci+sst+sans)/5f;  
 System.out.printf("The percentage is:%.4f\n",res);  
 }  
}

* 1. **Constants**

We can declare constant using final keyword.

Eg: final double pi = 3.14

**Some important programs:**

**Q2: ExponentOperations.java**

package com.company;  
  
public class ExponentOperations {  
 public static void main(String[] args) {  
 int a = (int)Math.pow(2,3);  
 System.out.println(a);  
 }  
}

**Q3: Taking character input**

package com.company;  
  
import java.util.Scanner;  
  
public class Tut7\_PracticeSet\_Q2 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.in);  
// way to take character input  
 char ch = sc.next().charAt(0);  
 int a = ch+8;  
 System.out.println("the grade is:"+(char)(a-8));  
 }  
}

**2 Operators and expressions**

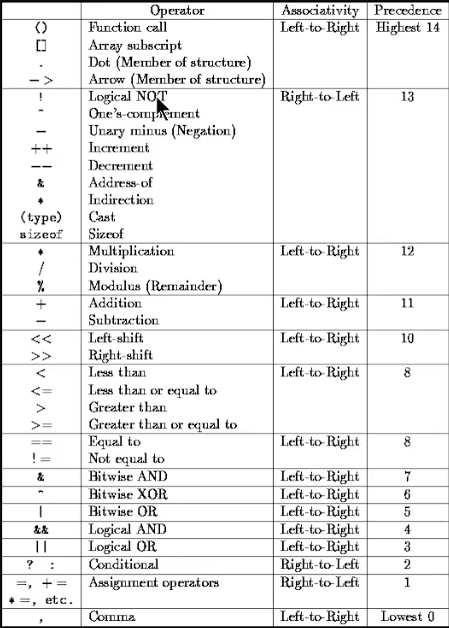
**2.1 Types of operators**

1. Arithmetic operators: +, -, \*, /, %, ++, --

Note: % operator works on float and double in Java

1. Assignment operators: =, += etc.
2. Comparison operators: ==, >=, <=
3. Logical operators: &&, ||, !
4. Bitwise operators: &, |

**2.2 Associativity and Precedence**

****

**Associativity:** It tells us the direction of execution of operators.

**Precedence:** It tells which operator needs to be applied first.

UCA – R to L

Other operators L to R

**2.3 Resulting Data type after an arithmetic operation**

It always converts to the bigger datatype

|  |  |
| --- | --- |
| **b:byte** | **f:float** |
| **s:short** | **d:double** |
| **i:integer** | **c:char** |
| **l:long** |  |

**Eg:**

R = b + s -> int

R = s + i -> int

R = l + f -> float

R = i + f -> float

R = c + i -> int

R = c + s ->int

R = l + d -> double

R = f + d -> double

**3 Strings**

String name = new String(“Harry”);

**Ways to print string in Java**

1. System.out.println()
2. System.out.print()
3. System.out.printf()
4. System.out.format()

**%d:** int, long

**%f:** float, double

**%c:** char

**%s:** string

Note: Strings are immutable and can’t be changed

**3.1 String methods**

String name = “Aditya”;

Modifying methods excluding length

1. name.length(): gives length of the string
2. name.toLowerCase(): converts string to lowercase and returns it
3. name.toUpperCase(): converts string to uppercase and returns it
4. name.trim(): returns string after removing leading and trailing spaces
5. name.substring()
   1. name.substring(int start): returns a substring from start index including the start
   2. name.substring(int start, int end): returns a substring from start index including the start until end index excluding the end
6. name.replace(‘r’,’p’): Returns a new string with ‘r’ replaced with p

Checking and informing methods

1. name.startsWith(“Ad”): returns true if name starts with “Ad” else false
2. name.endsWith(“ya”): returns true if name ends with “ya” else false
3. name.charAt(2): returns char at given index
4. name.indexOf(s): returns first occurrence of string s in name
5. name.indexOf(s,3): returns occurrence of string s starting from index 3
6. name.lastIndexOf(s): returns last occurrence of string s in name
7. name.lastIndexOf(s,3): returns last index of string s before index 3
8. name.equals(“Harry”): returns true if exact match
9. name.equalsIgnoreCase(“aditya”): case doesn’t matter

**3.2 Escape Seq characters**

eg: \n, \t, \’, \\ etc.