**Primitive data types -**

|  |  |
| --- | --- |
| **Data Type** | **Wrapper class** |
| byte | Byte |
| short | Short |
| int | Integer |
| long | Long |
| float | Float |
| boolean | Boolean |
| char | Character |

NOTE - . Wrapper classes help data type conversions

. Long and integer methods support unsigned operations

. var unsigned = Integer.parseUnsigned(“1000000”);

EX - String doubleValue = “156.5”;

Double doubleObj = Double.parseDouble(doubleValue); // Static method

var byteValue = doubleObj.byteValue();

var intValue = doubleObj.intValue();

var floatValue = doubleObj.floatValue();

var stringValue = doubleObj.toString();

**Typecasting-**

int i = 100;

long L = i ; // no problem as we are widening the type

int i = L ; // compilation error as lossy value

PROPER CONVERSION -

Syntax - datatype varName = (datatype) ConversionTypeVariableName;

int i = (int) L ; // no error but loss of data is possible

**Explicit and implicit typing-**

|  |  |
| --- | --- |
| EXPLICIT | IMPLICIT |
| int i = 4; | var i = 4;  infers the datatype from the assigned value |

**Char to String -**

|  |  |
| --- | --- |
| CHAR TO STRING | METHODS |
| // create a char array  char[ ] charArr = {‘a’,’b’,’h’,’i’}; | // to uppercase  char a = ‘a’;  char uppera = Character.toUpperCase(a); |
| // to make it a string  String s =new String(charArr); | // to lowercase  char a =’A’;  char lowera = Character.toLowerCase(a); |
| // back to char array  var charArray = String.toCharArray(); |  |

**Strings -**

1. Strings are instances of the class String.

STRINGBUILDER :

StringBuilder can be used to append more strings

EX -

StringBuilder sb = new StringBuilder(“Hello”);

sb.append(“ World”);

STRING FORMAT

var a =”abhi”;

var b =”shek”;

var c = 70.2;

var temp = “first name %s last name %s weight %f ”;

var output = String.format(temp,a,b,c);

Output -

first name abhi last name shek weight 70.2

COMPARE STRINGS:

var a =”abhi”;

String b =”abhi”;

a.equals(b); // returns true

a.equalsIgnoreCase(b); //ignores case to compare

SUBSTRING :

String s =”Abhishek”;

s.substring(0,4); //outputs Abhi

s.substring(4): // outputs shek

**Exception Handling -**

**//** can have multiple catch blocks

try {

int[] i = new int[5];

i[5] = 2;

} catch (Exception e) { // unknown exceptions

e.printStackTrace(); // trace which type of exception

System.out.println(e.getMessage());

}

**Classes and objects -**

1. Classes are like blueprints
2. Objects are items made out of these blueprints
3. Classes have fields and methods
4. A field is also known as class member

**OOPS CONCEPTS -** <https://drive.google.com/file/d/1sTqaLsBNNCpHE1G8sv_hWLKi-sXSh_q9/view?usp=sharing>

**Data Structures -**

ARRAYS -

int[ ] i = new int[5];

ARRAYLIST:

ArrayList<String> str = new ArrayList<>();

str.add(b);

str.add(0,a);

LISTS -

List<string> ls = new ArrayList<>();

ls.add();

LINKED LISTS -

LinkedList<String> ls = new LinkedList<>();

ls.add();

STACKS -

Stack sc = new Stack();

sc.push();

sc.pop();

QUEUES -

Queue<String> qu = new LinkedList<>();

qu.add();

qu.peek()

HASHMAPS -

Map<k,v> hm = new HashMap<>();

hm.put(“key”, value);

var getVal = hm.get(“key”);

var getAllKeys = hm.keySet(); // retrieves all keys