- Variable is store data
- We can just use any simple alpha numeric name to store data
- To a value 10 we have to use a variable x=10;
- Java is strongly typed language
- We must specify the type of data we store in side of a variable
- There are 8 fundamental datatypes in Java
- To store value 10 we can use int type
- We have to write statement like **int x=10;**
- Here int is datatype of 10, x is variable name and 10 value assigned to it.
- Whenever you want to access 10 you can use x
    - System.out.println(x);
    - This will print x value
- The 8 fundamental datatypes are called as primitive types
    - byte: 8 bit data
        - Range of byte is 2^{-7} to 2^{7} -1
        - -128 to 127
        - byte b=100;
    - short: 16 bit data
        - Range of short is 2^{-15} to 2^{15} -1
        - short s=3200;
    - int: 32 bit data
        - Range of short is 2^{-31} to 2^{31} -1
        - int i=3200;
    - long: 64 bit data
        - Range of short is 2^{-63} to 2^{63} -1
        - long ln=3200;
        - long ln1=3200000000000000000000L;
        - You need to suffix the value with l/L if the value exceeds int range
    - float: 32 bit data
        - We can store 6-7 floating points
        - float f=1.1f;
        - You need to suffix the value with f/F
    - double: 32 bit data
        - We can store 15-16 floating points
        - double d=1.111111111111111;
    - char: 16 bit
        - We can store single character using single quote
        - char c =’x’;
    - boolean: 8bit
        - we can assign true or false keywords
        - boolean b = true;
- All primitive types are keywords in java
- You should not use these names as variable/method/class….etc names.
- You cannot apply any method on primitive type because primitive types are keywords

- Java gave us wrapper classes which are wrapped around primitive types
- Because of this we can apply some methods on wrapper types
- For every primitive type there is a wrapper type available
    - byte: Byte class
    - short: Short Class
    - int: Integer class
    - long: Long class
    - float: Float class
    - double: Double class
    - char: Character class
    - boolean: Boolean class
- We can use these types for storing similar data like primitive
- There are two more datatypes given java for storing data
    - String : It is a class given by java
        - We can store group of characters using double quotes
    - Object: It is super class in java
        - A variable with Object type can accept any type of data
- Mostly we work with int, double, boolean, String, Object
- **Type Cast**
    - We have to use this when storing One type variable data in other type of variable
    - Implicit Casting
        - We have to use it when storing Smaller type larger type
        - Storing byte type variable to int type variable doesn't need to specify anything
        - The conversion will be done automatically
    - Explicit cast
        - We have to use it when storing Larger type to smaller type
        - Storing int type variable to byte type variable needs to be specified explicit cast
        - byte b = (byte) i;
        - Here b is byte type variable and i is int type variable
- Converting String to number
    - We have to use Integer.parseInt method to convert string to number
    - If we store numbers as strings we cannot perform calculations
    - Whatever we read from application will be in form of strings

```
public class C3TypeCast {

   public static void main(String[] args) {

         byte b = 100;
         int i = b; // implicit cast

         b = (byte) i; // Explicit cast

         int i1 = 130;

         byte b1 = (byte) i1;
```

```
        System.out.println(b1);

        String strN1 = "100";
        String strN2 = "200";
        System.out.println(strN1 + strN2);

        int n1 = Integer.parseInt(strN1);
        int n2 = Integer.parseInt(strN2);
        System.out.println(n1 + n2);
    }

}
```

- **Variables vs Constants**
    - Variable value varies
    - We can change the value of variable in other statements
    - If we create Constant then the value remain same
    - You cannot change in other statements
    - Use final keyword before variable name for creating constants
    - The value what we specify in beginning is final
    - final int x=10;
    - You cannot change value of x in other statements
- **Arrays**
    - Storing multiple values in a single variable using index number is called array
    - We need to define array with size
    - We need to use [ ] for defining array
    - Syntax for creating array for storing strings
        - DatatypeName [] variableName = new Datatype[numberOfValues];
        - DatatypeName variableName[] = new Datatype[numberOfValues];
    - How much size specified while defining array we can store only those number of values
    - Whatever the type we specify we can store only type data
    - If you give Object type then we can store any type of data
    - The array index starts from zero
    -