OBJECT™
TECHNOLOGIES

Language Basics

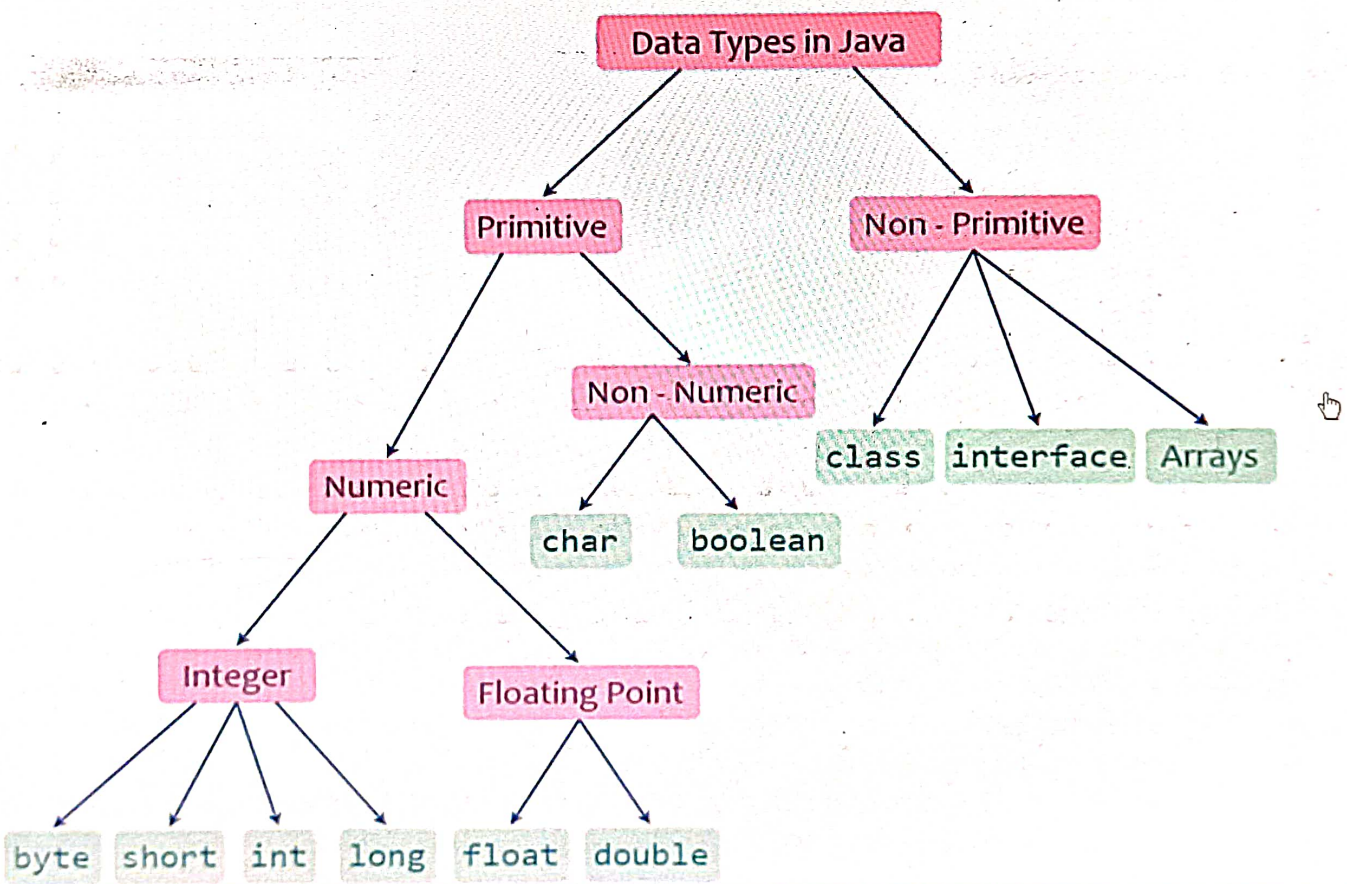
What will be covered

OBJECT TECHNOLOGIES

- Data types
- Operators
- Conditional statements
- Iterative Statements
- Wrapper classes
- Use of Arrays
- class String

Data Types

OBJECT TECHNOLOGIES



Data Types

OBJECT TECHNOLOGIES

- In java, data types are majorly divided into 2 categories - primitive or value type and non-primitive or reference type. Primitive data types hold the values where as non-primitive types refer to some objects. Non-primitive will not hold any values.
- In primitive types the further classification is done based on whether the data type holds the numric value or non-numeric value. In numeric we have different interger and floating types and in non-numeric char and boolean.
- Under non-primitive, majorly classes, interfaces, arrays are considered.
- Very important amongst primitive type is char and boolean. Numeric data types will always include range from negative numbers to positive numbers.
- In java, char data type occupies 2 bytes of space. Because it is not ASCII character but it is considered as UNICODE character. UNICODE is a set of characters from all different internationally recognized languages. It is not just English character set
- Other major difference is in size of boolean data type which occupies only one bit of memory to recognize whether the value is true or false.

Data Types

OBJECT TECHNOLOGIES

Type	Description	Default	Size	Example Literals
boolean	true or false	false	1 bit	`true, false`
byte	twos complement integer	0	8 bits	(none)
char	Unicode character	`\u0000`	16 bits	`'a', '\u0041', '\101', '\\', '\'', '\n', 'ß'`
short	twos complement integer	0	16 bits	(none)
int	twos complement integer	0	32 bits	`-2, -1, 0, 1, 2`
long	twos complement integer	0	64 bits	`-2L, -1L, 0L, 1L, 2L`
float	IEEE 754 floating point	0.0	32 bits	`1.23e100f, -1.23e-100f, .3f, 3.14F`
double	IEEE 754 floating point	0.0	64 bits	`1.23456e300d, -1.23456e-300d, 1e1d`

Operators

OBJECT TECHNOLOGIES

Operation Types	Operators	Examples
Arithmetic Operations	+, -, *, /, %, ++, --	A+B, A-B
Relational Operations	==, !=, >, <, >=, <=	A==B, A!=B
Logical Operations	&&, ||, !	(A==B) && (A<B)
Assignment Operations	=, +=, -=, *=, /=, %=	B=10, A+=20
Ternary Operations	(condition)? value if true : Value if false	X=(A<B)? 10 20
Bitwise Operations	&, |, ~, ^	A&B, A | B

Conditional Statements

OBJECT TECHNOLOGIES

Conditional statementa allows to execute a perticular block of code based on the result of condition. In java following conditional statements are allowed.

1. if statement
2. nested if statement
3. if-else statement
4. if-else-if statement
5. Switch Case Statemen

Conditional statements - Switch case

OBJECT TECHNOLOGIES

- The switch statement in Java is a multi branch statement. We use this in Java when we have multiple options to select. It executes particular option based on the value of an expression.
- Switch works with the byte, short, char, and int primitive data types. It also works with enumerated types, the String class, and a few special classes that wrap certain primitive types such as Character, Byte, Short, and Integer.
- Java switch case is a neat way to code for conditional flow, just like if-else conditions. Before Java 7, the only means to achieve string based conditional flow was using if-else conditions. But Java 7 has improved the switch case to support String also.

```
switch (color) {
case "blue":
        System.out.println("BLUE");
        break;
case "red":
        System.out.println("RED");
        break;
default:
        System.out.println("INVALID COLOR
}
```

Iterative Statements

OBJECT TECHNOLOGIES

The java programming language provides a set of iterative statements that are used to execute a statement or a block of statements repeatedly as long as the given condition is true. The iterative statements are also known as looping statements or repetitive statements. Java provides the following iterative statements.

1. while statement - Condition is checked at the beginning. If condition is false, no statement in the while loop gets executed
2. do-while statement - Conditio is checked at the time of exiting from loop. This loop gets executed atleast once
3. for statement - n for-statement, the execution begins with the initialization statement. After the initialization statement, it executes Condition. If the condition is evaluated to true, then the block of statements executed otherwise it terminates the for-statement. After the block of statements execution, the modification statement gets executed, followed by condition again.
4. for-each statement - It provides an approach to traverse through an array or collection in Java. The for-each statement also known as enhanced for statement. The for-each statement executes the block of statements for each element of the given array or collection. It is introduced from java version 5

Wrapper classes

OBJECT TECHNOLOGIES

- The wrapper class in Java provides the mechanism to convert primitive into object and object into primitive.
- In other words, wrapper classes provide a way to use primitive data types (int, char, short, byte, etc) as objects
- Need of wrapper classes
  - Wrapper Class will convert primitive data types into objects. The objects are necessary if we wish to modify the arguments passed into the method (because primitive types are passed by value).
  - The classes in java.util package handles only objects and hence wrapper classes help in this case also.
  - Data structures in the Collection framework such as ArrayList and Vector store only the objects (reference types) and not the primitive types.
  - The object is needed to support synchronization in multithreading.
- Note: Primitive types are more efficient than corresponding objects. Hence, when efficiency is the requirement, it is always recommended primitive types.

Wrapper classes

OBJECT TECHNOLOGIE

Primitive Data Type	Wrapper Class
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean

All numeric wrapper classes inherit from java.lang.Number

Wrapper classes

OBJECT TECHNOLOGIES

- Autoboxing
- The automatic conversion of primitive data types into its wrapper class objects is known as autoboxing.

```
int a = 15; // Primitive data type
Integer I = a; // Autoboxing will occur internally.
```

- Autounboxing
- The automatic conversion of wrapper class objects into its primitive data types is known as unboxing.

```
Integer a = new Integer(15); // Wrapper class object
int I = a;// Unboxing will occur internally.
```

Use of Arrays

OBJECT TECHNOLOGIES

- Java provides a data structure, the array, which stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type
- In Java all arrays are dynamically allocated using new operator. Every element in the array is recognized by using index which will start with 0. Maximum index available will be length-1
- Since arrays are objects in Java, we can find their length using the object property length.
- Declaring Array Variables
  - dataType[] arrayName;
  - e.g double[] data;
- Creating array instance
  - data = new Double[10];
  - Here data is an array which can store 10 double values

Use of Arrays

OBJECT TECHNOLOGIES

- Array declarion and creation can be clubbed in one statement
- Initializing array
  - e.g int[] age = {12, 4, 5, 2, 5};
  - we have created an array named age and initialized it with the values inside the curly brackets.
  - Note that we have not provided the size of the array. In this case, the Java compiler automatically specifies the size by counting the number of elements in the array (i.e. 5).
  - Alternatively, // declare an array
  - int[] age = new int[5];
  - // initialize array
  - age[0] = 12;
  - age[1] = 4;
  - age[2] = 5; ..

Use of Arrays

OBJECT TECHNOLOGIES

- Accessing array elements :
  - Array elements can be accessed using specific index value

    e.g. System.out.println("First Element: " + age[0]);
  - Using for loop

```
for(int i = 0; i < age.length; i++) {
    System.out.println(age[i]);
}
```

    we are using the length property of the array to get the size of the array.
  - Using for-each loop

```
for(int a : age) {
    System.out.println(a);
}
```

class String

OBJECT TECHNOLOGIES

- String is a sequence of characters, for e.g. “Hello” is a string of 5 characters. In java, string is an immutable object which means it is constant and can cannot be changed once it has been created.
- Strings can be created by assigning a String literal to a String instance:

  String str1 = "Welcome";
- String can be created using new keyword

  String str1 = new String("Welcome");
- String manipulation is one of the most common activities in computer programming. String class has a variety of methods for string manipulation.
- These functionalities include getting the character at perticular index, concatenating, replacing the character, trimming etc

class String

OBJECT TECHNOLOGIES

String method	Purpose
public char charAt(int index)	To get chracter at specified index
public String concat(String s)	Appening s at the end of source string
public boolean equalsIgnoreCase(String s)	Source string string given sstring compared without consideration of cases
public int compareTo(String s)	compares the two strings based on the Unicode value of each character
public indexOf(char ch)	Returns the index of first occurrence of the specified character
public String substring(int index)	returns the substring of the string.
public int length()	returns the length of the String
public String replace(char old, char new)	any occurrence of the char in the first argument is replaced by the char in the second argument
public boolean contains("searchString")	method returns true of target String is containing search String

Assignments

1. Write a program to check whether given number is odd or even
2. Declare 3 int values and display the greatest amongst the numbers.
3. Write a program to check whether the given number is prime or not
4. Write a program to find all the divisors of a given number e.g Divisors of 24 are 1,2,3,4,6,8,12,24
5. Write a program to calculate factorial of a given number
6. Write a program to print following pattern

```
5
5 4
5 4 3
5 4 3 2
5 4 3 2 1
```

7. Write program to print generate following pattern

```
1 1 1 1
2 2 2
3 3
4
```

8. Write program to print fibonacci series up to 10 terms

   e.g- 0 1 1 2 3 5 8 13 21 34

9. Write program to find sum of digit of given number upto single digit.(Declare any number with >3 digit)

   e.g. 2536= 2+5+3+6 =16

   16=1+6 =7

10. Print following pattern

```
    *
   * *
  * * *
 * * * *
* * * * *
```

11. Print following pattern

```
1
2 3
```

```
  * * * *
* * * * *
```

11. Print following pattern

```
1
2 3
4 5 6
7 8 9 10
```

12. Check whether the given number is armstrong or not? e.g no 153 is armstrong

$1^3+5^3+3^3 = 153$

13. Display the reverse of a given number.

14. Check whether the given number is perfect ot not? perfect number is a positive integer that is equal to the sum of its proper divisors e.g. 6 is perfect number becuase sum of it's divisors are equal to number itself

1 + 2 + 3 = 6

15. Print following pattern

```
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
```

16. Display the grade for the given percentage.

if the percentage is greater than or equal to 90, grade is "Excellent"
if the percentage is greater than or equal to 80, grade is "Very good"
if the percentage is greater than or equal to 70, grade is "Good"
if the percentage is greater than or equal to 60, grade is "Average"
if the percentage is greater than or equal to 40, grade is "Pass"
else grade is "Fail"

17. Display whether the given numbe is binary or not

18. Convert the given binary number into decimal and display.