Practical -3
Java classes and Objects

Java Classes/Objects

Everything in Java is associated with classes and objects, along with its attributes and methods.

For example: in real life, a car is an object. The car has attributes, such as weight and color, and methods, such as drive and brake.
A Class is a "blueprint" for creating objects.

Create a Class

Test.java
Create a class named "Main" with a variable x:

```
public class Main
{ int x = 5;

}
```

Create an Object

new operator is used to create an object of a class.
Instance variable and methods of a class can be accessed by dot operator

objectname.instance variable
objectname.instancemethod

each object have its own copy of the variables/methods

Create two objects of Main:

```
public class Test
{ int x = 5;
  public static void main(String[] args)
    {      Test myObj1 = new Test(); // Object 1
           Test myObj2 = new Test(); // Object 2
           System.out.println(myObj1.x);
           System.out.println(myObj2.x);
    }
}
```

Java Constructors

A constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes.

Example Create a constructor:

BoxDemo.java

```
 class Box
{   double width;
    double height;
    double depth;
// compute and return volume
  double volume()
     { return width * height * depth; }
          // sets dimensions of box
   //Constructor of Box class
   Box()
        { System.out.println("Constructing Box");
             width = 10;
              height = 10;
               depth = 10;
         }

}

public class BoxDemo
    {  public static void main(String args[])
            { // declare, allocate, and initialize Box objects
                    Box mybox1 = new Box();
                   Box mybox2 = new Box();
                  double vol;
                           // get volume of first box
                  vol = mybox1.volume();
                  System.out.println("Volume is " + vol);
                          // get volume of second
                  box vol = mybox2.volume();
                  System.out.println("Volume is " + vol);
         }

 }
```

Parametrized Constructor

it is used to pass the arguments/Parameter to the constructor. It is used to pass the values to the object at the time of creation of it.

```
class BoxDemo3
 {
    double width;
    double height;
    double depth; // This is the constructor for Box.
```

```
BoxDemo3(double w, double h, double d)
      {   width = w;
           height = h;
           depth = d;
       }

     // compute and return volume
       double volume()
         { return width * height * depth;
          }

     public static void main(String args[])
      { // declare, allocate, and initialize Box objects
           Box mybox1 = new Box(10, 20, 15);
           Box mybox2 = new Box(3, 6, 9);
          double vol;
                // get volume of first box
          vol = mybox1.volume();
          System.out.println("Volume is " + vol);
         // get volume of second box
          vol = mybox2.volume();
          System.out.println("Volume is " + vol);
      }
}
```

Java String Methods

charAt()	Returns the character at the specified index (position)	char
compareTo()	Compares two strings lexico graphically	int
concat()	Appends a string to the end of another string	String
contains()	Checks whether a string contains a sequence of characters	boolean
contentEquals()	Checks whether a string contains the exact same sequence of characters of the specified CharSequence or StringBuffer	boolean
copyValueOf()	Returns a String that represents the characters of the character array	String
endsWith()	Checks whether a string ends with the specified character(s)	boolean
equals()	Compares two strings. Returns true if the strings are equal, and false if not	boolean
format()	Returns a formatted string using the specified locale, format string, and arguments	String
indexOf()	Returns the position of the first found occurrence of specified characters in a string	int
isEmpty()	Checks whether a string is empty or not	boolean

length()	Returns the length of a specified string	int
replace()	Searches a string for a specified value, and returns a new string where the specified values are replaced	String
split()	Splits a string into an array of substrings	String[]
substring()	Returns a new string which is the substring of a specified string	String
toCharArray()	Converts this string to a new character array	char[]
toLowerCase()	Converts a string to lower case letters	String
toString()	Returns the value of a String object	String
toUpperCase()	Converts a string to upper case letters	String
trim()	Removes whitespace from both ends of a string	String

- **charAt()**

```
String myStr = "Hello";
char result = myStr.charAt(0);
System.out.println(result);
```

output : H

- **CompareTo()**

Returns 0 if both the string are same lexically

```
String myStr1 = "Hello";
String myStr2 = "Hello";
byte z =(myStr1.compareTo(myStr2); // Returns 0 because they are equal
System.out.println(z)  // 0
```

- **Contact()**

The concat() method appends (concatenate) a string to the end of another string.

Syntax :

String concat(String *string2*)

e.g

```
String firstName = "John ";
String lastName = "Doe";
String str= firstName.concat(lastName);
System.out.println(str);
```

- **Contains()**

boolean contains(CharSequence *chars*)

```
String myStr = "Hello";
System.out.println(myStr.contains("Hel"));   // true
System.out.println(myStr.contains("e"));     // true
System.out.println(myStr.contains("Hi"));    // false
```

- **equals()**
- equals() method compares two strings, and returns true if the strings are equal, and false if not.

```
String myStr1 = "Hello";
String myStr2 = "Hello";
String myStr3 = "Another String";
System.out.println(myStr1.equals(myStr2)); // Returns true because they are equal
```

System.out.println(myStr1.equals(myStr3)); // false

- **indexOf()**
  - An int value, representing the index of the first occurrence of the character in the string, or -1 if it never occurs

    public int indexOf(String *str*)
    public int indexOf(String *str*, int *fromIndex*)

    String myStr = "Hello planet earth, you are a great planet.";
    System.out.println(myStr.indexOf("e", 11));
    Output : 13

- **Length()**

```
String txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
System.out.println(txt.length());
```

- **substring()**

Syntax :

```
public String substring(int startIndex) ;
public String substring(int startIndex, int endIndex) ;
```

e.g

```
String s1="java with fun";
System.out.println(s1.substring(2,4));    //returns  va
System.out.println(s1.substring(2));      //returns vatpoint
```

- **replace() :**

The replace() method searches a string for a specified character, and returns a new string where the specified character(s) are replaced.

Syntax

String replace(char *searchChar*, char *newChar*)

e.g

```
String myStr = "Hello";
System.out.println(myStr.replace('l', 'p'));   // Output Heppo
```

Exercise

1	Create a class Car . Car have attribute like modelname, color,manufacturing year. Class contain a method to display() : to display the detail of the car. Can have constuructor which passes the value of car attribute. Create a java program which create a 3 car object and display the detail of each object.
2	Create a class called Calculator, takes two number. And have method for arithmatic operations add(), sub(), mul(), div() which returns the result. Write a java code which create object of Calculator class and call the mehtod for arithmatic operations.
3	Write a Java program which calculates the area of Circle, Triangle and Rectangle. Create three different classes which call the Constructors of Rectangle, Triangle and Circle by passing parameters to the constructors.

	Create a main method in class named Shape which provides the user a menu through which the user can calculate the area of the respected shape using the area() method of that class .
4	Write a java program which takes string values from the user. Make use of different constructors of String Class to initialize string and demonstrate the implementation of methods of String class.