# Java OOP 10128

## Introduction, Python to Java

Pini shlomi

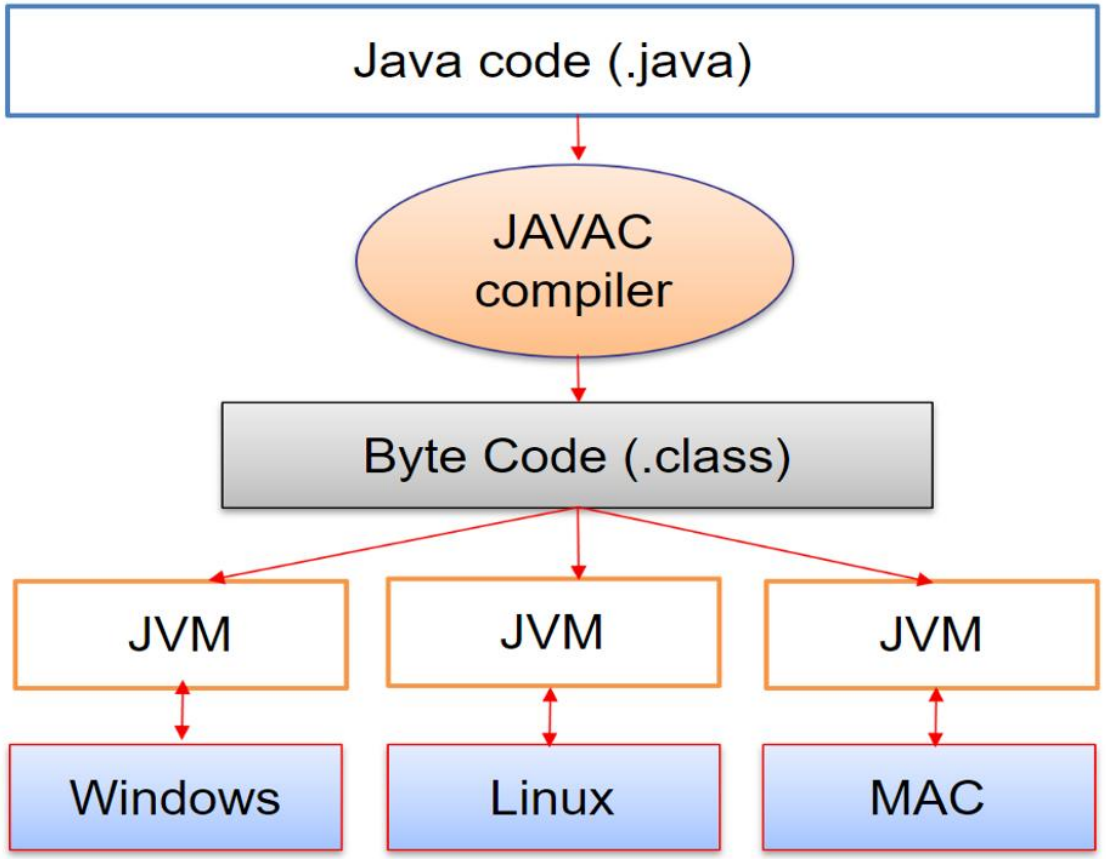
# Object Oriented Programing

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Data is encapsulated within objects, with properties (attributes) and behaviors (methods).
- OOP concepts enable writing modular, reusable, and maintainable code
  - Abstraction - simple things to represent complexity
  - Encapsulation - keeping fields within a class private, then providing access to those fields via public methods.
  - Inheritance - share some of the attributes of existing classes.
  - Polymorphism - same word to mean different things in different contexts.

# How Java is portable?

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992




Java class is written in Unicode characters

Java compiler convert these Unicode characters into byte code

Java byte code can only be understandable by JVM

JVM is native code and specific to OS

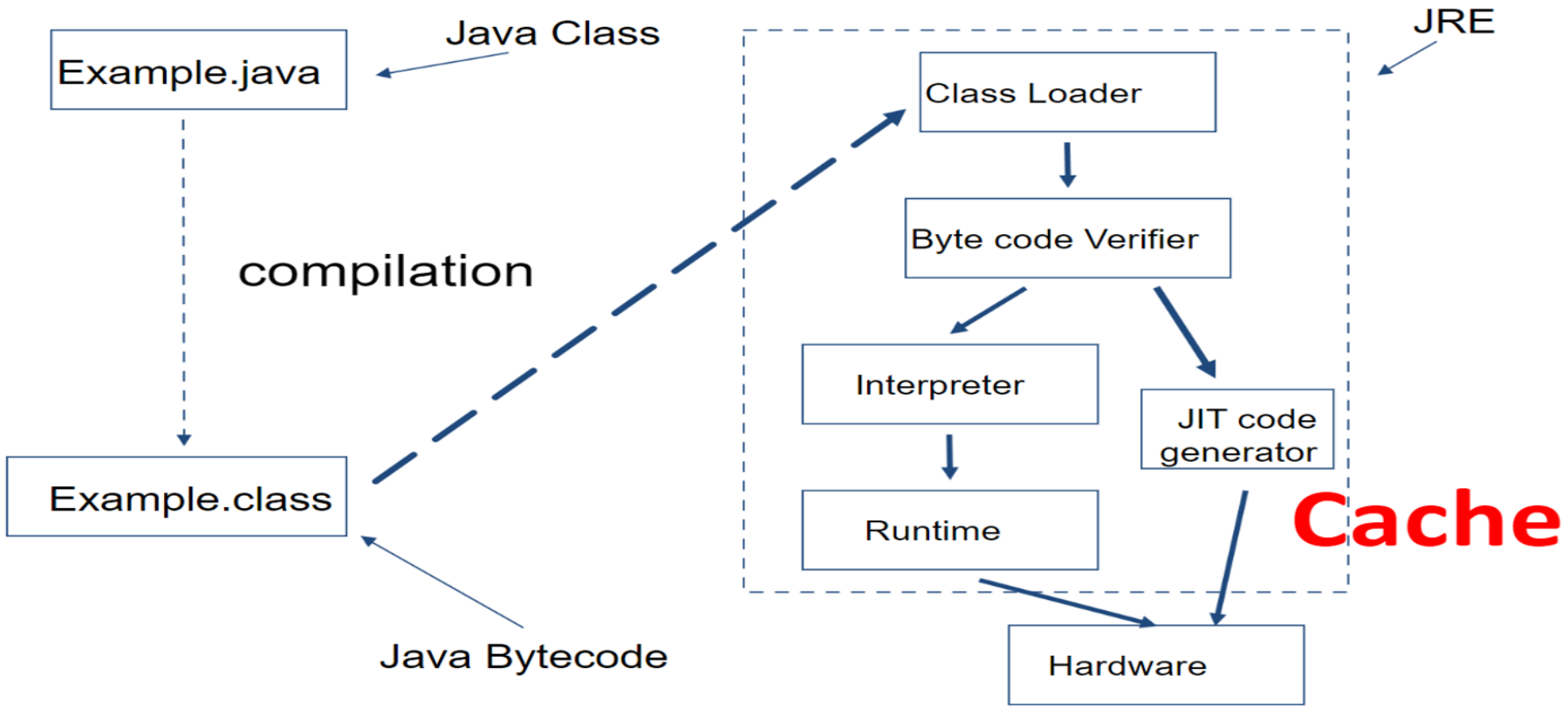
# Java portability

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- All java code is cross platform
- Rich set of libraries classes implementation for all platforms
- Compiled once run anywhere
- Primitive have definite size
- Unicode character set, 16-bit superset of ASCII

# Java Runtime Environment
# JIT – (Just-In-Time) compiler

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992



# Stack and Heap

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

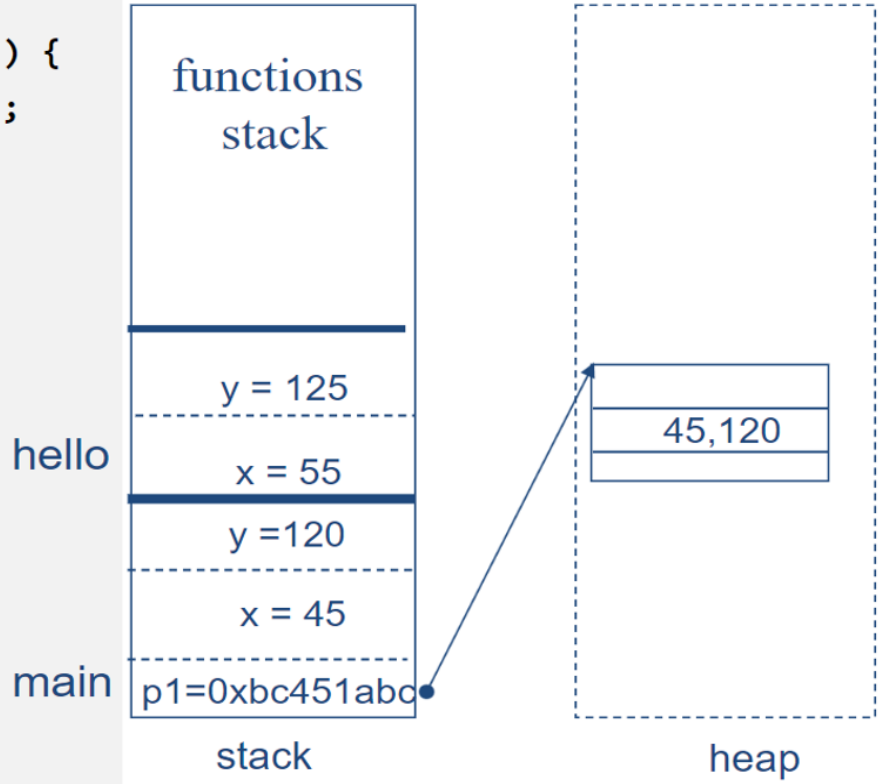
- Object data saved in heap memory
- Object reference saved in stack memory

Visualize code

```
public class Test {
        public static void main(String args[]) {
                Point p1 = new Point(45,120);
                int x = 45;
                int y = 120;
                hello();
                int z = 500;

        }
        public static void hello() {
                int x = 55;
                int y = 125;
                System.out.println("Hello");

        }
```



# Syntax

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- All commands must be inside class.
- Class name always start with an uppercase first letter
- Case-sensitive
- Class name must be same as filename
- Every program must contain main method
- Curly braces { } marks the beginning and the end of a block of code
- Each code statement must end with a semicolon (;).
- Comments
  - Single-line comments start with two forward slashes (//).
  - Multi-line comments start with /* and ends with */.

# Variables

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

Variables are containers for storing data values.

- Declaring or Creating Variables

  type variableName; Or type variableName = value; => `int myNum = 15;`

- Names can contain letters, digits, underscores, and dollar signs
- Names must begin with a letter
- Names should start with a lowercase letter, and cannot contain whitespace
- Names can also begin with $ and _
- Names are case-sensitive ("myVar" and "myvar" are different variables)
- Java keywords cannot be used as names

# Java Keywords

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

| | | | | |
|---|---|---|---|---|
| abstarct | continue | for | new | switch |
| assert | default | goto | package | synchronized |
| boolean | do | if | private | this |
| break | double | implements | protected | throw |
| byte | else | import | public | throws |
| case | enum | instanceof | return | transient |
| catch | extends | int | short | try |
| char | final | interface | static | void |
| class | finally | long | strictfp | volatile |
| const | float | native | super | while |

# Primitive Data Types

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

| Data Type | Size | Description |
|---|---|---|
| byte | 1 byte | Stores whole numbers from -128 to 127 |
| short | 2 bytes | Stores whole numbers from -32,768 to 32,767 |
| int | 4 bytes | Stores whole numbers from -2,147,483,648 to 2,147,483,647 |
| long | 8 bytes | Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
| float | 4 bytes | Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits |
| double | 8 bytes | Stores fractional numbers. Sufficient for storing 15 decimal digits |
| boolean | 1 bit | Stores true or false values |
| char | 2 bytes | Stores a single character/letter or ASCII values |

# Java wrapper classes - immutable

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

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

- Autoboxing - conversion primitive to wrapper class.
- Unboxing - conversion wrapper class to primitive.

# Non-Primitive Data Types

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Called also reference types because they refer to objects.

The main difference :

| Non-Primitive Data Types | Primitive Data Types |
|---|---|
| not defined by Java (except for String | predefined |
| call methods to perform certain operations | cannot |
| can be null | has always a value |
| starts with an uppercase letter. | starts with a lowercase letter |

# Variable Examples

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
// Integer (whole number)
    int myNum = 5;
// Floating point number
    float myFloatNum = 5.99f;
// Character
    char myLetter = 'D’;
// Boolean
    boolean myBool = true;
// String
    String myText = "Hello";
```

# Java Type Casting

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

Widening Casting (automatically) - converting a smaller type to a larger type size

byte -> short -> char -> int -> long -> float -> double

Narrowing Casting (manually) - converting a larger type to a smaller size type

double -> float -> long -> int -> char -> short -> byte

# Output - print () function

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Print everything as strings.
- Prints strings in between double quotes or variables
- Not appends a newline to output.
- To print with a newline, use `println()`.

```
System.out.println("Hello World!");

System.out.println(3 + 3);

System.out.println(firstName + " " + lastName);
```

# Input

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

input() function is to read input from the standard input (the keyboard, by default)

It accepts all user input as a string and converting them depend the methods.

```
Scanner s = new Scanner(System.in);
System.out.println("Enter integer number: ");
int int_num = s.nextInt();
System.out.println("int number is " + int_num);
s.close();
```

```
Enter integer number:
25
int number is 25
```

# Input

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
Scanner s = new Scanner(System.in);
System.out.println("Enter your name: ");
String st = s.next();
System.out.println("Enter char: ");
char ch = s.next().charAt(0);
System.out.println("your name is " + st);
System.out.println("char is " + ch);
s.close();
```

```
Enter your name:
pini
Enter char:
S
your name is pini
char is S
```

# Arithmetic operators

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

| Operator | Name | Example |
|---|---|---|
| + | Addition | x + y |
| - | Subtraction | x - y |
| * | Multiplication | x * y |
| / | Division | x / y |
| % | Modulus | x % y |
| ++ | Increment | x++ |
| -- | Decrement | x-- |

# Assignment operators

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

| Operator | Example | Same As |
|---|---|---|
| = | x = 5 | x = 5 |
| += | x += 3 | x = x + 3 |
| -= | x -= 3 | x = x - 3 |
| *= | x *= 3 | x = x * 3 |
| /= | x /= 3 | x = x / 3 |
| %= | x %= 3 | x = x % 3 |

# Comparison Operators

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

| Operator | Name | Example |
|---|---|---|
| == | Equal | x == y |
| != | Not equal | x != y |
| > | Greater than | x > y |
| < | Less than | x < y |
| >= | Greater than or equal to | x >= y |
| <= | Less than or equal to | x <= y |

# Logical Operators

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

| Operator | Name | Example |
|---|---|---|
| && | And | x && y |
| \|\| | Or | x \|\| y |
| ! | Not | !(x > y) |

# Condition

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
if (condition) {
    // block of code to be executed if the
    // condition is true
}else {
    // block of code to be executed if the
    // condition is false
}
```

Shorthand If...Else

variable = condition ? expressionTrue : expressionFalse;

```
String result = time < 18 ? "Good day." : "Good evening.";
```

# String

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Immutable sequences of ordered characters
- Enclosed in double "quotes".

```
String txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
System.out.println("The length of the txt string is: "
      + txt.length());

String txt = "Hello World";
System.out.println(txt.toUpperCase());
      // Outputs "HELLO WORLD"

System.out.println(txt.toLowerCase());
      // Outputs "hello world"
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 1 :

ציינו את ההבדלים בתחביר בין Python ל- Java

בתחומים הבאים :

- הצהרת משתנים.
- פלט למסך.
- קריאה של קלט מהמשתמש.

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 2 :

מהן מחלקות ה- wrapper ב- Java וכיצד הן קשורות לסוגי הנתונים הפרימיטיביים?

רמז : Unboxing ,Autoboxing

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 3 :

מהן ארבעת עקרונות ה- OOP וכיצד כל אחד מהם בא לידי ביטוי ב-Java?

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 4 :

נתון קטע קוד ב- Python כתבו קוד מקביל ב-java.

```
x = 10
y = 20
print(f"Sum: {x + y}")
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 5 :

כתבו תוכנית ב- Java שקולטת מספר מהמשתמש ומדפיסה אם הוא זוגי או אי-זוגי.

# Switch

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
switch(expression) {
    case x:
        // code block
        break;
    case y:
        // code block
        break;
    default:
        // code block
}
```

# Traditional Switch

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
switch (errorCode) {
    case 404, 405:
        System.out.println("Not found!");
        System.out.println("Please correct your request.");
        break;
    case 418:
        System.out.println("I am a teapot!");
        break;
    case 500:
        System.out.println("Internal server error!");
        break;
    default:
        System.out.println("Unknown code!");
}
```

# Enhanced Switch

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
switch (errorCode) {
    case 404, 405 -> {
        System.out.println("Not found!");
        System.out.println("Please correct your request.");
    }
    case 418 -> System.out.println("I am a teapot!");
    case 500 -> System.out.println("Internal server error!");
    default -> System.out.println("Unknown code!");
}
```

# While Loop

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
while (condition) {
    // code block to be executed
}
```

```
int i = 0;
while (i < 5) {
    System.out.println(i);
    i++;
}
```

# Do While Loop

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

Will always be executed at least once,
even if the condition is false.

```
int i = 0;
do {
    System.out.println(i);
    i++;
} while (i < 5);
```

# For Loop

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
for (statement 1; statement 2; statement 3) {
      // code block to be executed
}
```

Statement 1 : is executed (one time) before the execution of the code block.

Statement 2 : defines the condition for executing the code block.

Statement 3 : is executed (every time) after the code block has been executed.

```
for (int i = 0; i < 5; i++) {
     System.out.println(i);
}
```

# For-Each Loop

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

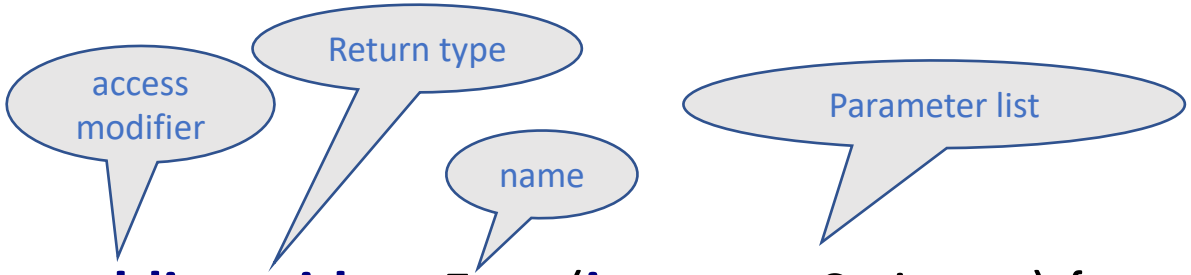
```
for (type variableName : collection) {
    // code block to be executed
}
```

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (String i : cars) {
    System.out.println(i);
}
```

# Function

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
<access-modifier> <return-type> <function-name>(<parameter-list>)
        throws <exception-list> {
        // function body
}
```




```
public  void myFunc(int num, String st) {

        // function body

}
```

# Function Parameters

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
public static void main(String[] args) {
    myMethod("Yael", 35);
    myMethod("Mor", 28);
    myMethod("Dan", 31);
}
```

Calling function with arguments

Function Parameters

```
static void myMethod(String firstname, int age) {
    System.out.println(firstname + " is " + age);
}
```

```
Yael is 35
Mor is 28
Dan is 31
```

# Function Overloading

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

Define multiple functions with the same name but different parameter lists or types

```
public static void main(String[] args) {
        int sum1 = add(5, 10);
        System.out.println("Sum of two integers: " + sum1);
        int sum2 = add(5, 10, 15);
        System.out.println("Sum of three integers: " + sum2);
}
public static int add(int a, int b) {
   return a + b;
}
public static int add(int a, int b, int c) {
   return a + b + c;
}
```

```
Sum of two integers: 15
Sum of three integers: 30
```

# main Function

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Entry point of Java program
- Name must be ‘main’.
- args parameter for command-line arguments
- return type is void

```
public static void main(String[] args) {
    // Method body
}
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

# תרגיל 6:

כתבו פונקציה ב- Java שקולטת מספר שלם מהמשתמש ומדפיסה את חישוב העצרת שלו.

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

# תרגיל 7:

כתבו תוכנית ובה תפריט של 5 אפשרויות וכן אפשרות נוספת ליציאה, בכל אפשרות הפעילו פונקציה אחרת ובה הדפיסו את מספר האפשרות שהמשתמש בחר ואם בחר לצאת הדפיסו "להתראות".

קלטו מהמשתמש את האפשרות והציגו בהתאם את התוצאה על המסך.

# Array

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

Store multiple values in a single variable

Declaration:

type [] varibaleName;

```
int[] arr1;
arr1 = new int[4];
int[] arr2 = {1,2,3};
```

# Array

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Access the Elements of an Array

  Array indexes start with 0

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars[0]);  // Outputs Volvo
```

- Change an Array Element

```
cars[0] = "Opel";
System.out.println(cars[0]);  // Now outputs Opel instead of Volvo
```

- Array Length

```
System.out.println(cars.length);  // Outputs 4
```

# Array

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Arrays Loop with for

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (int i = 0; i < cars.length; i++) {
    System.out.println(cars[i]);
}
```

- Arrays Loop with for-each

```
for (String car : cars) {
    System.out.println(car);
}
```

# Multi-Dimensional Arrays

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
System.out.println(myNumbers[1][2]); // Outputs 7
myNumbers[1][2] = 9;
System.out.println(myNumbers[1][2]); // Outputs 9 instead of 7

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

# Package

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

- Used for organizing classes into namespaces.
- provide a way to group related classes together.
- provide access control mechanisms through their visibility modifiers (public, protected, package-private, private).
- Classes within the same package can access each other's members with default (package-private) or no visibility modifiers.

```
package com.example.app;

public class Main {
        // class definition
}
```

# Package

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

## Summary of Access modifier.

| Access Modifier | Visibility within the Package | Visibility outside the Package | Visibility within Subclasses | Example |
|---|---|---|---|---|
| public | Visible | Visible | Visible | public class MyClass { ... } |
| protected | Visible | Not Visible | Visible | protected void myMethod() { ... } |
| Package-private | Visible | Not Visible | Not Visible | void myMethod() { ... } |
| private | Not Visible | Not Visible | Not Visible | private int myField; |

# Package - Example

אפקה
המכללה האקדמית
להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

```
package com.example.superclass;
public class Superclass {
    protected void superClassMethod() {
        System.out.println("Superclass method");
    }
}
package com.example.subclass;
public class Subclass extends Superclass {
    public void subclassMethod() {
        System.out.println("Subclass method");
        superClassMethod(); // Accessing protected method from superclass
    }
}
```

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

תרגיל 8 :

כתוב את הפונקציה getMinToStart() המקבלת מערך numbers של מספרים שלמים. הפונקציה תשנה את סדר איברי המערך numbers כך שבסוף ריצתה, numbers[0] יכיל את המספר הקטן ביותר מבין המספרים ב- numbers. הסדר של שאר המספרים במערך אינו משמעותי.

דוגמאות:

עבור המערך numbers = [-3,5,-2,1,-7,8,9,3],

פלט אפשרי הוא [-7,-3,5,-2,1,3,8,9].

עבור המערך numbers = [7,9,2,0,0,1],

פלט אפשרי הוא [0,7,9,2,0,1].

אפקה
המכללה האקדמית להנדסה בתל אביב
פיני שלומי מהנדס תוכנה 054-4636992

# תרגיל 9 :

כתוב פונקציה המקבלת מערך של מספרים שלמים חיוביים ושליליים.
כל איבר במערך ייוצג ע"י שורה בציור, ויוצגו כוכביות כערך האיבר:

- אם הערך שלילי הכוכביות יוצגו משמאל לנקודת המרכז, בצמוד לקו האמצע
- אם הערך חיובי הכוכביות יוצגו מימין לנקודת המרכז, בצמוד לקו האמצע
- אם הערך הוא 0 לא יוצגו כוכביות

```
The array is: [0, -6, -4, 9, 8]
      |
******|
  ****|
      |*********
      |********
```

```
The array is: [8, 4, -3, -3, -6]
      |********
      |****
   ***|
   ***|
******|
```

```
The array is: [-3, -3, -9, 9, 8]
      ***|
      ***|
*********|
         |*********
         |********
```