Week 1

- A computer consists of:
  - Hardware: Physical devices of computer system
  - Software: Programs that run on computers
    - A program is a set of instructions written in a high programming language (such as C/C++ or Java) to do a specific task.

- The main components of a computer system are:

1. Input unit (mouse, keyboard)
2. Output unit (printer, monitor, audio speakers)
3. Memory unit (retains input data, calculated data and instructions)
4. Central processing unit (CPU) which consists of:

4.1 Control unit (supervises the operation of other devices)
4.2 Arithmetic and Logic unit (ALU) (performs calculations)

5. Secondary storage (hard drives, floppy drives)

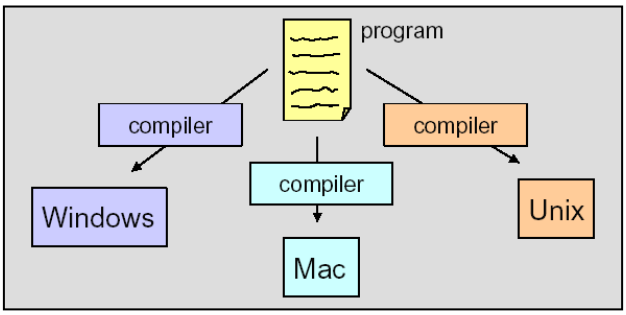
- Solving a program goes through the following main steps:

1. Analysis: Outline the solution requirements
2. Design an appropriate algorithm or a flowchart.
3. Code the solution in a high programming language (such as Java)
4. Compile the code into machine language. Verify that the program works:
   4.1 If there is an error (Syntax error), correct it by going to step 3.
   4.2 If there is no error, proceed to step 5.
5. Run the program. Verify the results:
   5.1 If the output does not give the required results (Run time & logic error), go to step 1.
   5.2 If the output matches the required result, you are done.

- A program consists of the following:
  - Input: Data to begin with to solve the program.
  - Output: The target of the problem. This is the expected result.
    - *Nouns in the problem statement suggest input and output data.*
  - Processing: This is the set of instructions that drive from the input to the output.
    - *Verbs in the problem suggest the processing steps.*

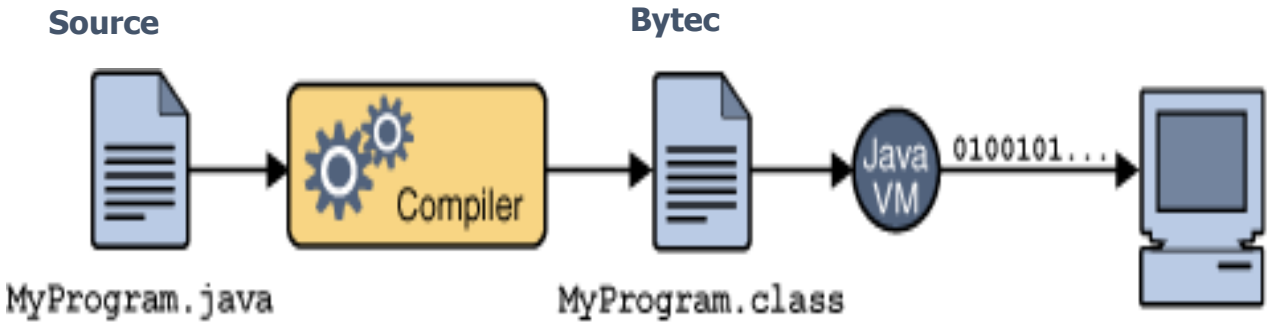
- Computers do not understand programs written in high programming languages (source code) such as C++ and Java
- Therefore, programs must be converted into machine code that the computer can understand and execute
- A software that translates a source code from a high-level programming language into machine code is called a compiler

- The compiler is a software that checks the correctness of the source code according to the language rules.
- If the compiler produces an error, this is called a syntax error.
- Translates the source code into a machine code if no errors were found.
- Machine code depends on the computer hardware: we say that the compiled version is platform-dependent.
    - For example, a program compiled on a machine that works under the Windows operating system, cannot run on another machine that works under the MAC operating system.
    - In this case, the program should be re-compiled under the MAC operating system.
- However, Java is characterized by being platform-independent. In other words, a Java program that is compiled under the Windows OS can run under the MAC OS without being re-compiled.



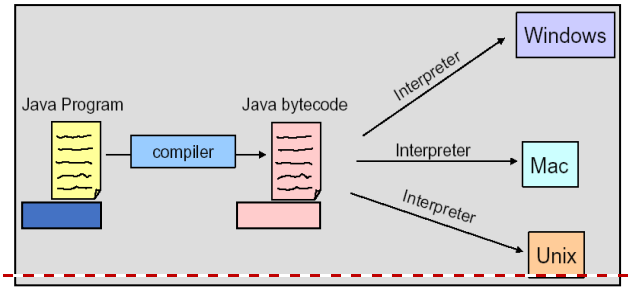
Platform-Dependent Compilation

- The Java compiler translates the source code (with extension ".java") into a bytecode (with extension ".class") rather than machine code.
- Then, a bytecode is converted into machine code using a Java Interpreter.

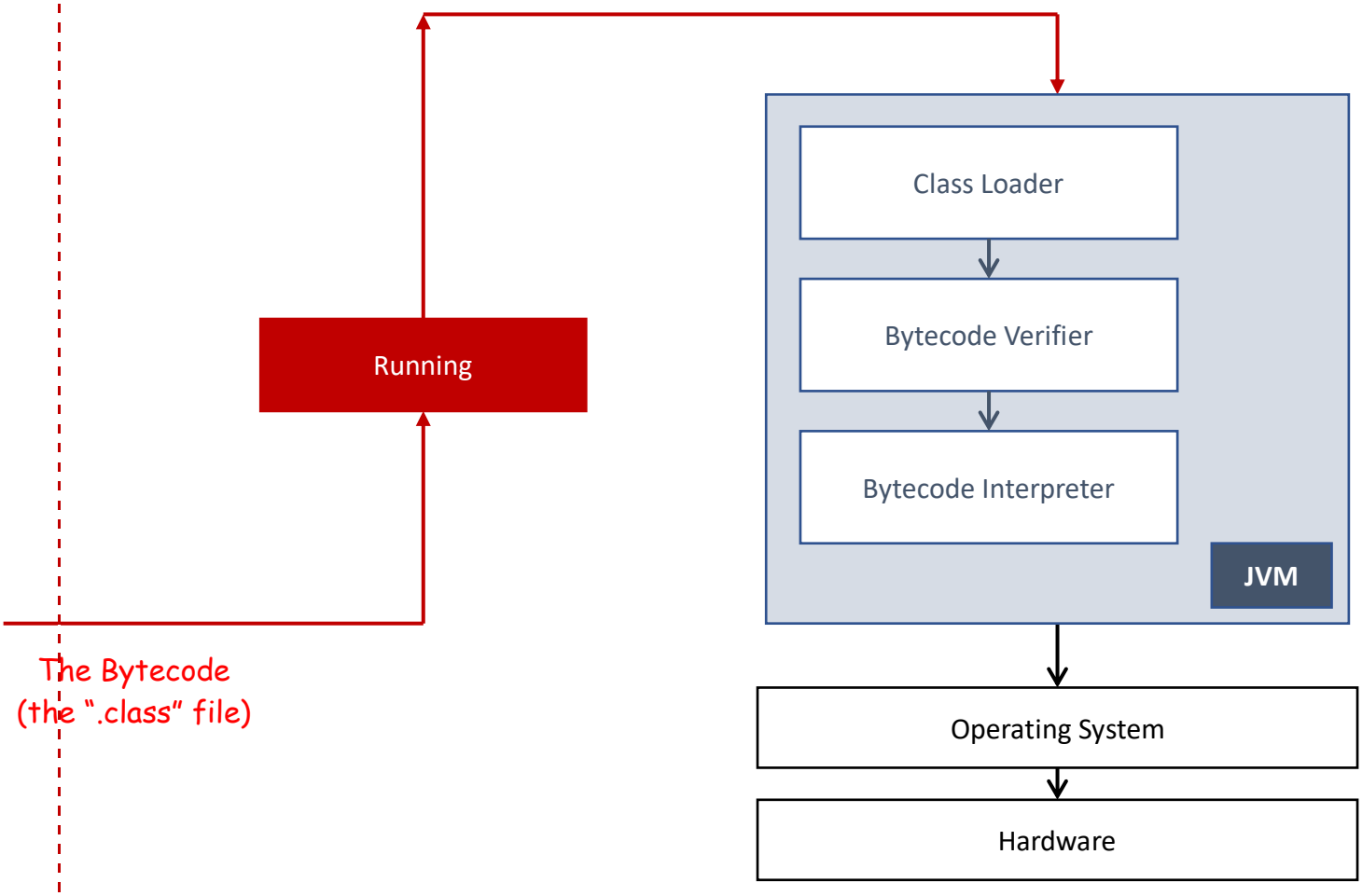


Java VM or JVM = Java Virtual Machine

The same bytecode is run on any computer installed with a Java Interpreter.

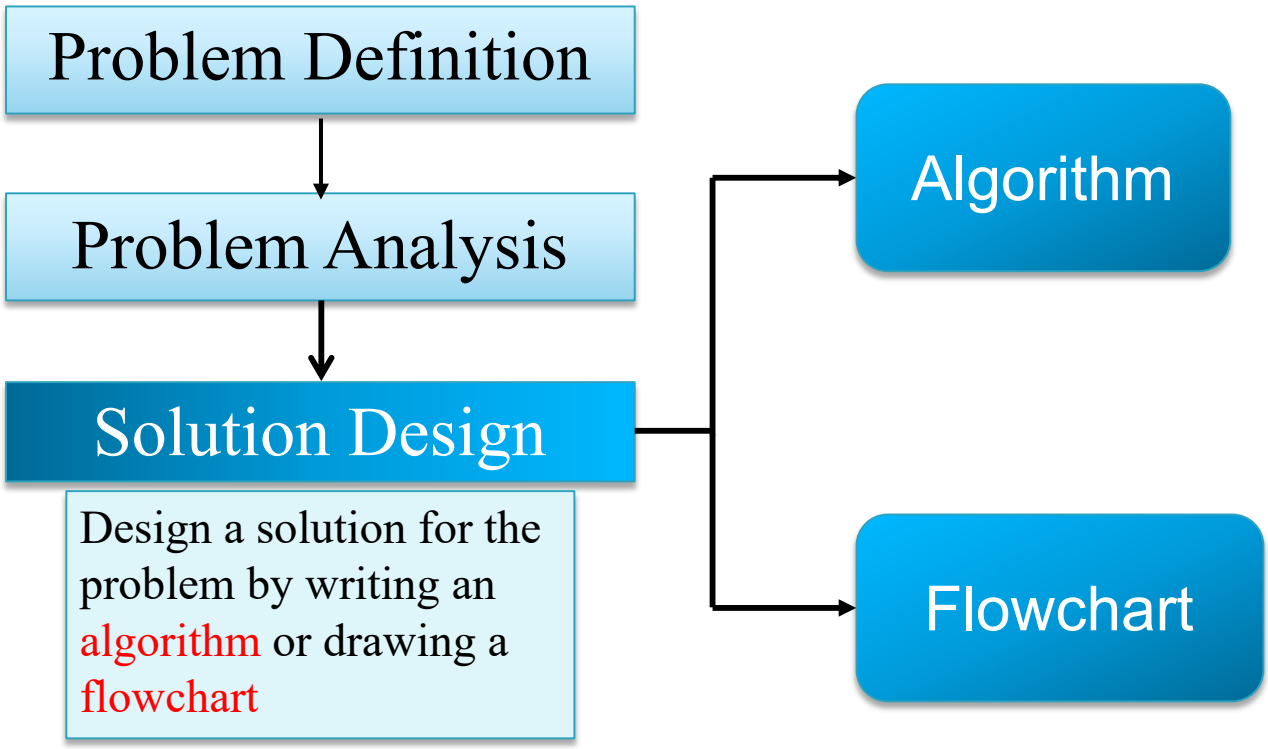


- Java Virtual Machine (JVM): A hypothetical computer developed to make Java programs machine independent (i.e. run on many different types of computer platforms).
- A JVM consists of the following components:
  - The class loader: stores the bytecodes into the computer memory
  - Bytecode verifier: ensures that the bytecodes do not violate security requirements
  - Bytecode interpreter: translates the bytecode into machine language



- Testing
  - After program compilation, the programmer needs to make sure that the output of the program matches the expected results.
  - Two types of errors may be encountered
    - Logical Errors: The program runs but provides wrong output.
    - Runtime errors: The program stops running suddenly. This happens when the program asks the OS to execute a non-accepted statement (such as division by zero).
- Debugging
  - When an error is encountered, it should be found, understood, and corrected.

- The basic steps to solve a problem are:
  - Problem Definition: Define the main problem
  - Problem Analysis:
    - Identify the inputs
    - Identify the outputs
    - Identify the processing operations
  - Solution Design:
    - Identify the detailed steps that the computer will follow to reach the expected result



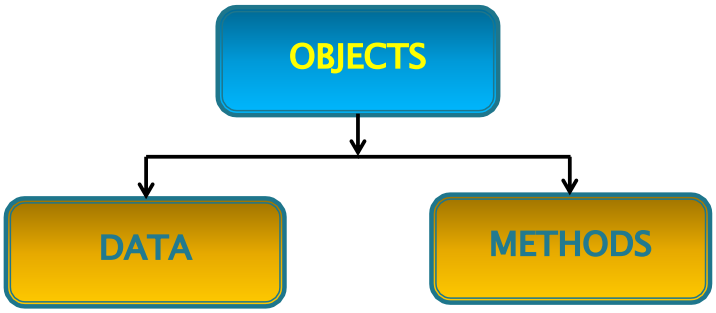
Flowchart:

- A flowchart is the graphical representation of the algorithm.
- The flowchart illustrates the previously developed algorithm in more details.
- These details make the algorithm very close to the code implementation in any high-level programming language.
- Worth to mention, that both algorithm and flowchart are independent of the used programming language.
- The flowchart uses standard symbols to illustrate each action. These symbols are shown in the next slide.

Week 2

Two programming approaches are known:

- **The Structured Programming**
  - Also known as modular programming.
  - The problem is divided into smaller sub-problems (modules).
  - Each sub-problem is then analyzed and solved.
  - The solutions of all sub-problems are then combined to solve the overall problem.
  - Examples of such programming model languages include Pascal, Fortran and C.
- **The Object-Oriented Programming**
  - Identify the components of the problem. These are called objects.
  - For each object, identify the relevant data & operations (methods) to be performed on that data.
  - Define the relationship between each object and the other.
  - Examples of programming languages that follow such model are C++ and Java.

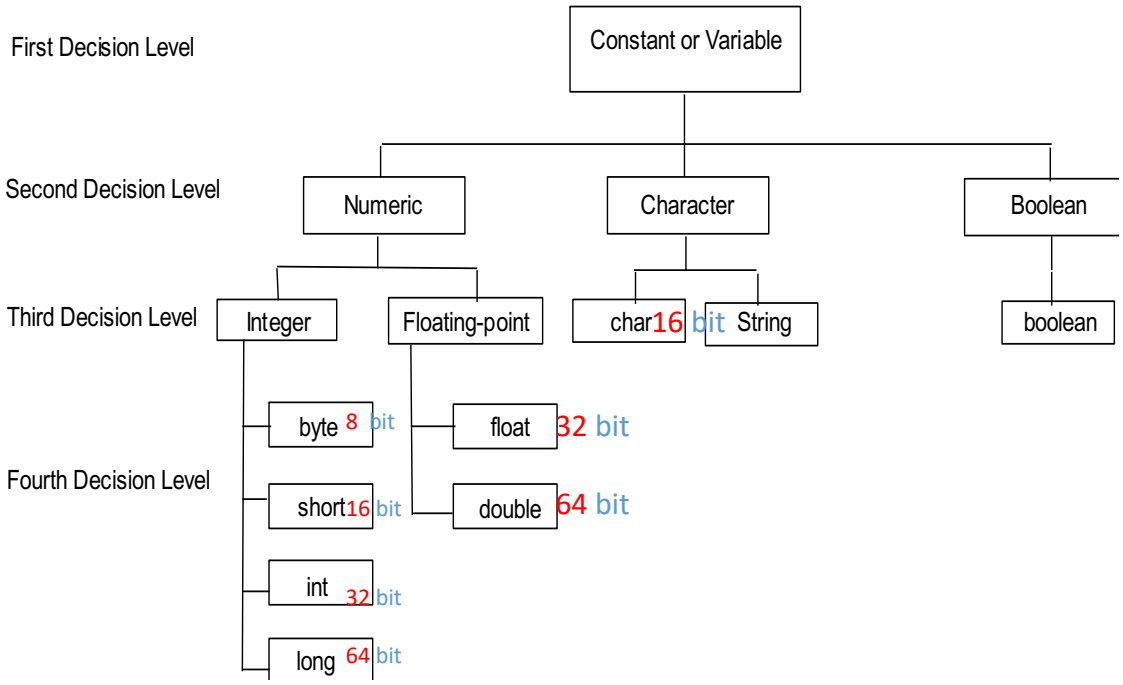


- **Rules for identifiers' names include:**
  - May consist only of:
    - Letters (a – z or A – Z),
    - Digits (0 – 9),
    - Underscore (_),
    - Dollar sign ($)
  - Should NOT begin with a digit
  - Not a reserved word:
    - These are some words used in the Java language.
    - They are interpreted by the compiler to do a specific thing.
    - Examples of reserved words include: public, class, void, etc…

, NuIdentifier names are case sensitive: numbermber, NUMBER represent three different identifiers.

Data type

- The data type defines what kinds of values a space memory is allowed to store.
- All values stored in the same space memory should be of the same data type.
- All constants and variables used in a Java program must be defined prior to their use in the program.



- The state of the space memory is the current value (data) stored in the space memory.
- The state of the space memory:
- May be changed.
- In this case the space memory is called variable.
- Cannot be changed.
- In this case the space memory is called constant.

- Constants:
  - All uppercase, separating words within a multiword identifier with the underscore symbol, _.
- Variables
  - All lowercase.
  - Capitalizing the first letter of each word in a multiword identifier, except for the first word.

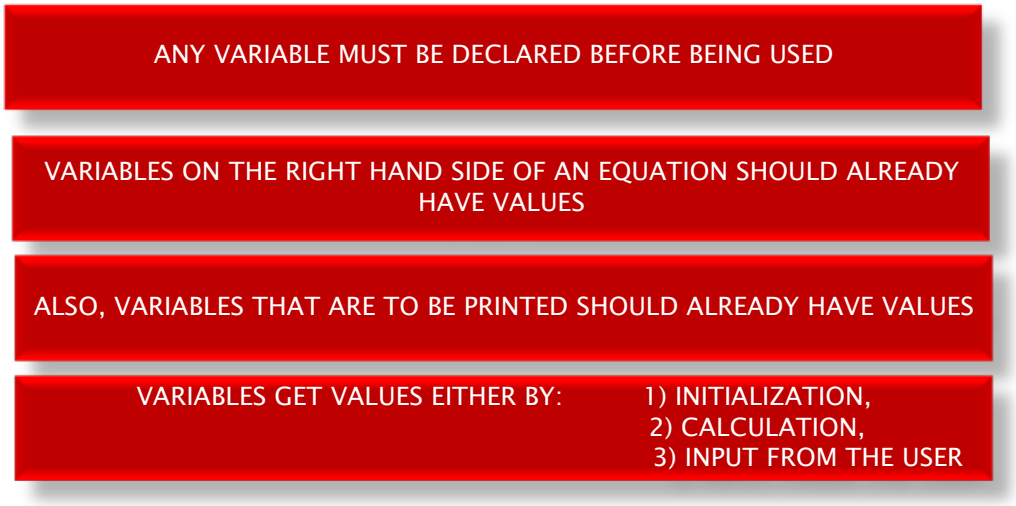
- Declaration allocates appropriate memory space to identifiers based on their types.
- Any identifier must be declared before being used in the program.
- The declaration of a variable means allocating a space memory which state (value) may change.
- The declaration of a constant means allocating a space memory which state (value) cannot change.

- A variable may be declared:
  - With initial value.
  - Without initial value.

- In Java, double is the default type of a floating-point number.

- When using float literals, the number should be written as shown below; otherwise, the compiler would give an error message (syntax error) :

```
float x=5.33f;
float length=12.33f, width= 6.333f, radius=0.3f;
```



Week 3

- There are five types of operators
  - Assignment
  - Arithmetic
    - Addition (+)
    - Subtraction (-)
    - Multiplication (*)
    - Division (/)
    - Modulus (%)
  - Increment/Decrement
  - Relational
    - == equal to
    - != not equal to
    - < less than
    - <= less than or equal
    - > greater than
    - >= greater than or equal
  - Logical
    - ! not
    - && and
    - || or

Parenthesis ( )	inside-out
Increment (++), Decrement (--)	from left to right
* / %	from left to right
+ -	from left to right
< > <= >=	from left to right
== !=	from left to right
&&	from left to right
||	from left to right
= += -= *= /= %=	from left to right

TYPE	METHOD NAME
Scanner input = new Scanner (System.in);	
byte b;	b = input.nextByte();
short sh;	sh = input.nextShort();
long lg;	lg = input.nextLong();
float flt;	flt = input.nextFloat();
char ch;	ch = input.next().charAt(0);

- nextDouble() is a method associated with the class Scanner. It accepts values that can be interpreted as a double type.
  - Integers may be interpreted as a double type.
  - A .0 is added to the integer.
  - If the next input cannot be interpreted as a double, then the program terminates with a run-time error message indicating an input mismatch.
  - Examples of invalid double include 24w5 or 2e10.

Week 4

- Escape sequences allow you to control the output.
- All escape sequences start with a backslash \ character.
- The following table shows some of the most commonly used escape sequences:

Syntax	Escape Sequence	Description
\n	New line	Cursor moves to the beginning of the next line.
\r	Return	Cursor moves to the start of the current line.
\t	Tab	Cursor moves to the next tab stop.
\b	Backspace	Cursor moves one space to the left.
\\	Backslash	Backslash is printed.
\’	Single quote	Single quotation is printed.
\”	Double quotes	Double quote is printed

- The argumentList is a list of one or more arguments: constant values, variables, or expressions.
- If the argumentList has more than one argument, then the arguments are separated by commas.

- The default output of floating-point numbers is:
  - up to 6 decimal places for float values, and
  - up to 15 decimal places for double values.

➢ The following are other format specifiers:

Specifier	Description
%d	The result is formatted as a (decimal) integer
%f	The result is formatted as a decimal floating-number
%c	The result is a Unicode character
%s	The result is a string
%e	The result is formatted as a scientific notation
%n	Line separator
%%	Prints %

- By default, the output is right-justified for all primitive data types.

```
String name = "aly";
System.out.printf ("First name = %6s%n", name);
```

```
1 First name = ~~~aly
2 _
```

- To print it left-justified, precede the width with a – sign:

```
String name = "aly";
System.out.printf ("First name = %-6s%n", name);
```

```
1 First name = aly~~~
2 _
```

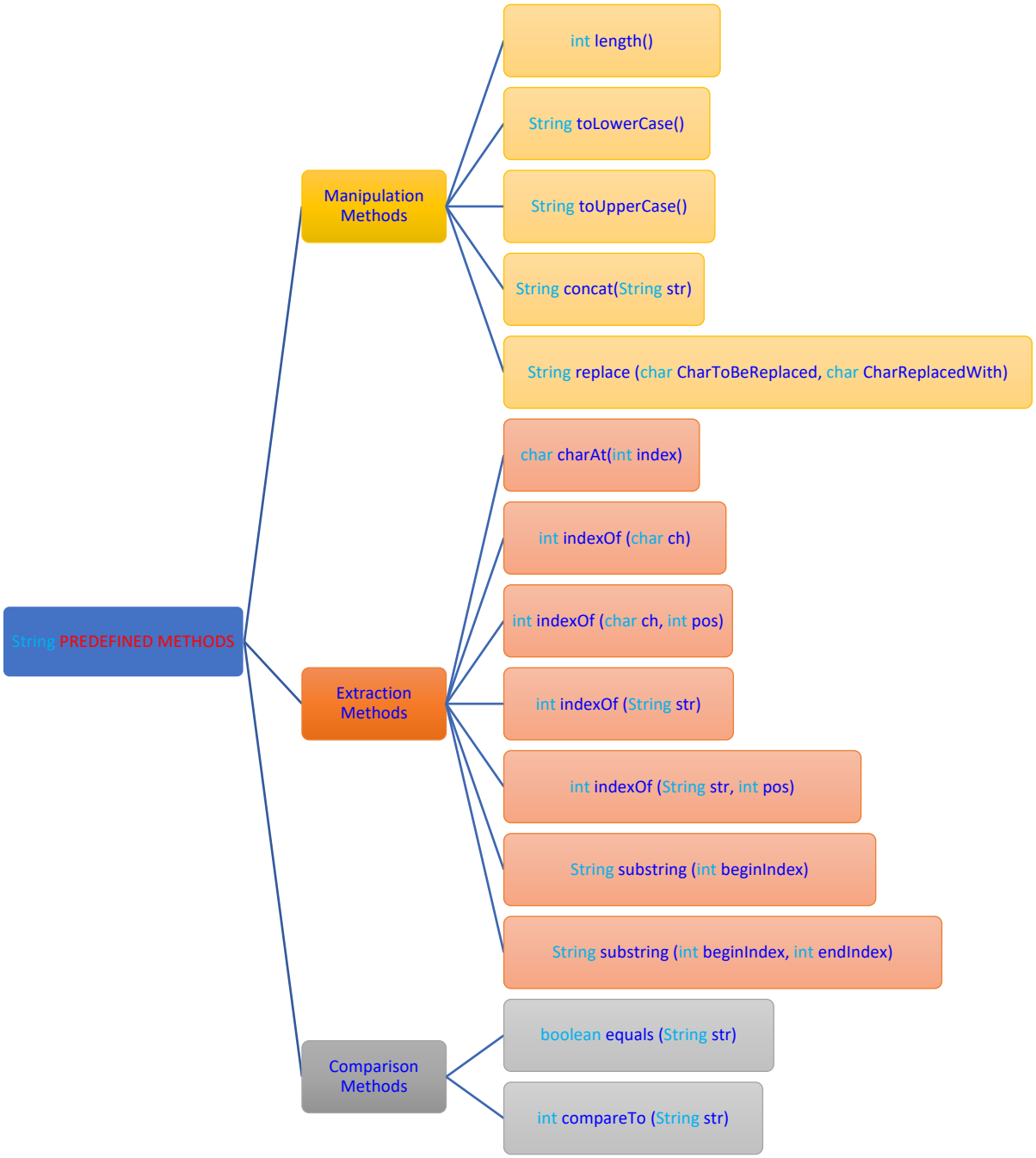
```
int year = 2015;
System.out.printf ("Academic year = %-6d-%6d", year, ++year);
```

```
1 Academic year = 2015~~-~~2016_
```

Week 5

- The class String contains many predefined methods.

- The dot (.) is called member access operator.
- Not all methods need parameters. Others may need more than one parameter.
- Parameters have predefined types that we should follow.
- In case the method has more than one parameter, the order is important.
- When a method returns a value of a specific type, then the result should be stored in a variable of the same type.

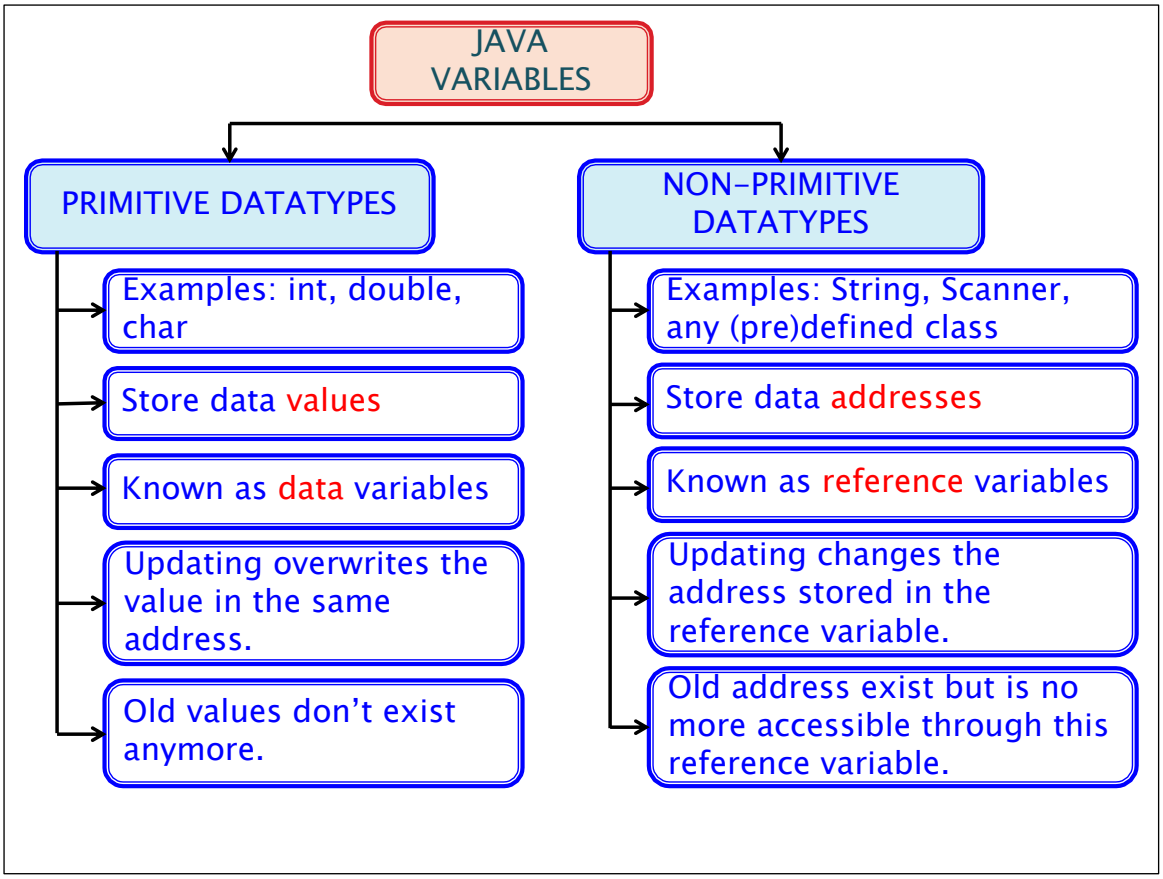

String PREDEFINED METHODS
Manipulation Methods
int length()
String toLowerCase()
String toUpperCase()
String concat(String str)
String replace (char CharToBeReplaced, char CharReplacedWith)
Extraction Methods
char charAt(int index)
int indexOf (char ch)
int indexOf (char ch, int pos)
int indexOf (String str)
int indexOf (String str, int pos)
String substring (int beginIndex)
String substring (int beginIndex, int endIndex)
Comparison Methods
boolean equals (String str)
int compareTo (String str)

- A String is a sequence of zero or more characters.
- In Java, a String is enclosed between "double quotation".
- Students' names, Universities' names, Countries' names are stored in a String.

- Every character in a String has a specific position.
- The position of the first character in the String is zero.
- The length of a String is the number of included characters.

- A String that contains no characters is called a null string or an empty string. This is written as "".

The length of the empty (null) string is zero.



Week 6

➢ **Validating user input is a vital issue in programming. The programmer should take into consideration all possible values entered by the user.**

- **In switch Value can't be double or float.**

- The switch statement is an elegant way to apply multiple selections. It is less confusing.
- However, the programmer is forced sometimes to use the nested if. Examples of such cases include:
  - If the selection involves a range of values.
    - Example: if (score >= 60) && (score < 70)
  - If the selector's type is double or float.
    - Example: if (salary == 5000.0)

➢ **CONDITIONAL OPERATOR '?' Also known as the ternary operator.**

Variable = (LogicalExpression1) ? Expression2: Expression3;

If expression1 = true, then the result of the condition is expression2. Otherwise, the result of the condition is expression3.

Week 7

➢ **The repetition or Looping need 3 Key elements to work correctly :**

- **Counter: to count how many times the loop was executed**
- **Condition: to test if the counter reached the desired number of loops / or a specific value**
- **Update: to update the counter**

- **The statement or block of statements in the loop will execute until the logical expression become false**

- **Statements must change value of expression to false.**
- **A loop that continues to execute endlessly is called an infinite loop (expression is always true).**

  - Infinite loops should be avoided. They are considered a "logical error".

  - Sometimes, we don't know how many times we want to repeat the loop.

  - In this case, the for statement cannot be used.

  - Therefore, the program is designed to repeat the loop until a special value is met.
  - This special value is called sentinel.
  - The value of the sentinel should be carefully selected, so that it cannot be by anyway a part of the data.

  - A random number is generated using the pre-defined method random().
  - random() belongs to the pre-defined class Math.

  - So, it is written as Math.random().

  - The Math class belongs to the package java.lang.*.
  - Math.random() returns a value of type double greater than or equal to zero and less than 1.0. ((Math.random() >= 0.0) && (Math.random() < 1.0)).

  - If you want to generate a random value of type int, use type casting as follows:
    num = (int)(Math.random()) ➔ ((num >=0) && (num < 1))

  - If you want to generate a random value of type int that ranges between 0 (inclusive) and 100 (exclusive), do the following:
    num = (int)(Math.random() * 100) ➔ ((num >=0) && (num < 100))

  - If you want to generate a random value of type int that ranges between 0 (inclusive) and 1000 (exclusive), do the following:
    num = (int)(Math.random() * 1000) ➔ ((num >=0) && (num < 1000))

  - If you want to generate a random value of type int that ranges between 0 (inclusive) and 1000 (exclusive), do the following:
    num = (int)(Math.random() * 1000) ➔ ((num >=0) && (num < 1000))

Week 8

- The statements in the loop body of a do…while are executed at least once.

Displaying a menu for the user and taking action according to the user's input is better implemented using do…while.

- The do…while statement is convenient to use to validate the user input.

If you know the number of required iterations ➔ use the for statement.

If you don't know the number of required iterations, and the loop may not execute at all ➔ use the while statement

If you don't know the number of required iterations, and the loop should execute at least once ➔ use the do…while statement

The break statement took you out of the block to next statement after the block.

The continue statement skip the statements in the loop and start the loop again.

Each loop (inner and outer) has its own counter.

Week 9

An array is a collection of two or more adjacent memory cells called array elements.

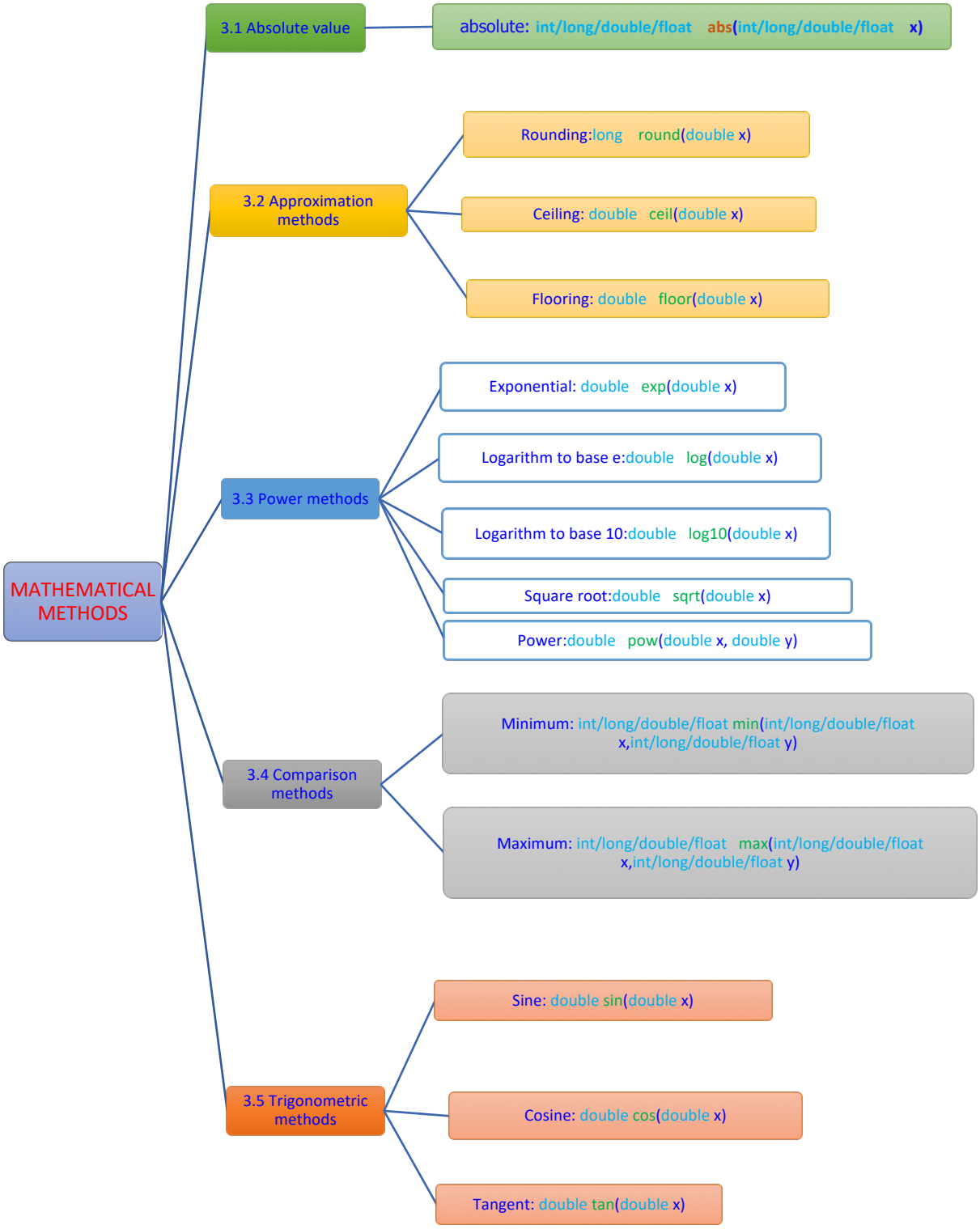
The size of the array specified during instantiation cannot be changed.

Two arrays are considered equal if:

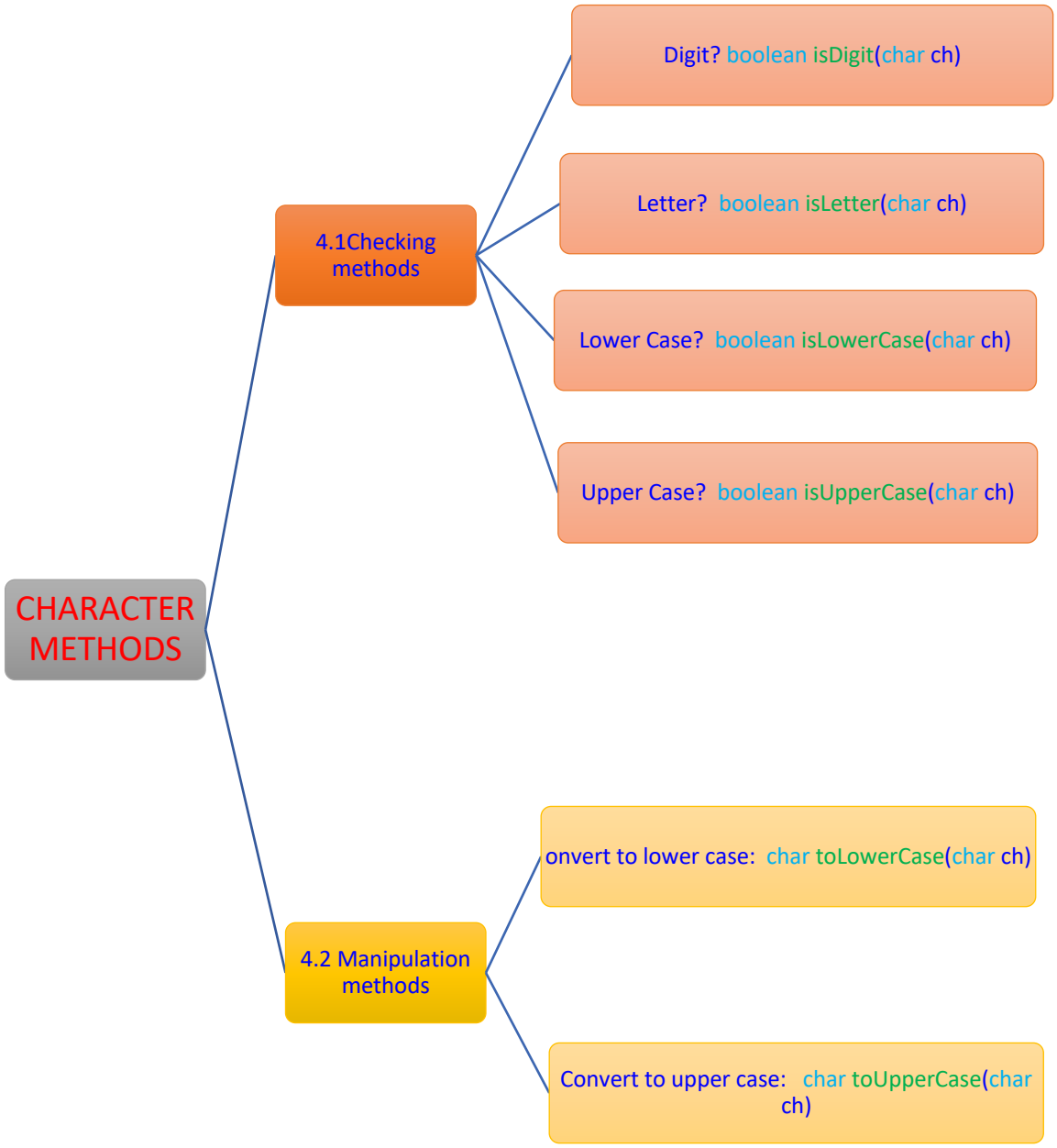
1. They have the same size, and
2. All corresponding elements in both arrays are equal.

Week 10

Mathematical methods belong to class Math.



Character manipulation methods belong to class Character.



- The returned value may be variable, constant value or expression.
- The type of the returned expression is the same as the method's type.
- The return statement is the last line executed in the method. After that, control is transferred back to the caller method (main for example).
- A return statement can return one value only.

- Variables declared inside the method are known as local variables.

	void	value-returning
return data?	No	Yes
Parameter list	May include from zero to any number of parameters	May include from zero to any number of parameters
Calling	Stand-alone statement in the caller method	Used in the same way as variables in the caller method

	Formal Parameters	Local Variables	Actual Parameters
Defined where?	Method header	Inside the method	Caller method
Used where?	Inside the method	Inside the method. Also, declared inside the method.	Values replace formal parameters of the called method
Value	Actual parameters pass their values to the formal parameters	Are given values inside the method	Should have values before being used to call a method.

- Make sure the following conditions are satisfied between the caller and called methods:
  - The number of parameters is equal
  - The parameter order is relevant (important)
  - The corresponding parameters types are the same
  - Don't write the types of the actual parameters in the caller method
  - Don't write [] for arrays in the caller method
- Make sure of the following in the value-returning methods:
  - A value is returned
  - The returned value has the same data type as the method
  - All paths are considered to return a value
  - Remember that the return statement is the last to execute in the method
  - Remember that the return statement passes a single value to the caller method

- When a method is called, the value of the actual parameter is copied into the corresponding formal parameter.

	Primitive	Arrays	String
Actual Parameter	Value	Address	Address
Connection between Actual & Formal Parameters	Ends as the actual parameters copy their values to the formal parameters	Is kept even after the called method finishes execution	Is kept UNTIL the String is updated inside the method.
Updates in method reflected on actual parameter?	No	Yes	No
How to transfer a value from the called to the caller	Use the return statement for non-void methods only	No need to do anything, since memory is updated directly	Use a method of type String, and return its address in a return statement. An example is given in the next slide. (same as primitive)

- Java provides the concept of variable length parameter list to simplify rewriting the method multiple times.
- A method can have at most one variable length formal parameter

➢ The following method headers are incorrect:

```
public static void myMethod (double … list1, int … list2)
```

- Because there are two variable length formal parameters.

```
public static void myMethod (double … list1, int number, String name)
```

- Because the variable length formal parameter should be the last parameter in the list.

```
public static void myMethod (double[] … list1)
```

- No parenthesis [] should be written with the variable length formal parameter.

Week 12

- What is OOP "object-oriented programming"?
  - OOP is a programming paradigm based on three simple core concepts:
    - Classes:
      - Are "**types**" of "things" that we can talk about.
    - Objects:
      - Are "**instances**" or examples of classes.
    - Message passing:
      - Objects **interact** with each other by passing messages to ask each other to carry out their methods (behaviours).

OOP advantages are:

- ☐ Easy to **understand**
- ☐ **Natural** partitioning of the problem
- ☐ Systems can be developed more **rapidly** and at a lower cost
- ☐ Directly related to **reality** - reduces the semantic gap between reality and models
- ☐ More **flexible** and resilient to change - allows local modification to models

Week 13

- An attribute has a name unique within the class.

- There are two types of attributes:
  - Class attributes (static variables)
    - Independent of any object and their values are shared by all objects of the class.
  - Instance attributes
    - Dependent to the objects and their values are associated with and accessed through objects.

- A **class attribute** is in one fixed location in memory.

- **Constructor method:** to initiate an object by sending its values as parameters
- **Accessor method (getters):** to get information about an object
- **Mutator methods (setters):**to mutate (change) an object's state
- **Printing method:** to specify how to print an object