

CS202: IT Workshop

Java

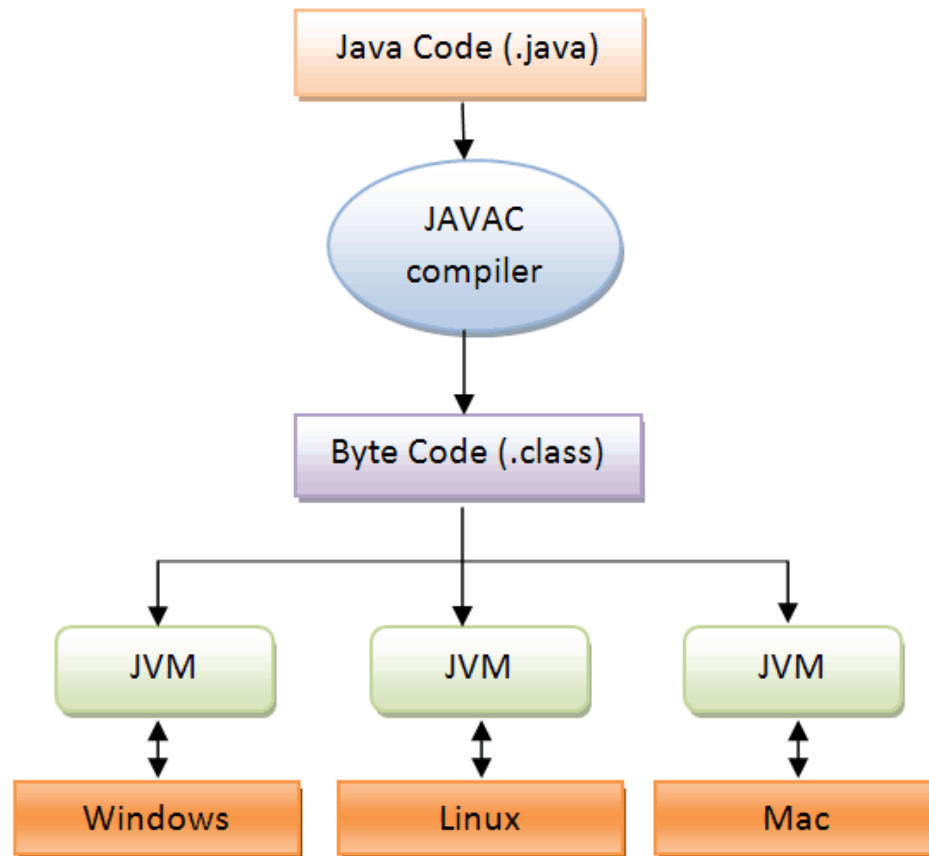
Reading from console, Basic programming constructs

Ref:

1. Harvey Deitel, Paul Deitel: Java How to Program, 9/e, Prentice Hall India.
2. <https://docs.oracle.com/en/java/>



Last class summary



- ❑ Bytecode is platform independent.
- ❑ JVM loads .class file, verifies, executes on the host machine with the help of host OS.

[Suggestion: Read details about JVM from Internet if interested]

Variables in Java

- ❑ Variable stores information
- ❑ Every variable has its name, type, size, value
- ❑ Java is a **statically-typed** programming language
(All the variables must be declared with its type before their use)
- ❑ Java supports both **primitive** and **reference** types of variables
- ❑ Primitive types are reserved words and written in lower case letters

Name	Description	Size (in bits)
boolean	true or false	1
byte	two's complement integer	8
character	Unicode character	16
short	two's complement integer	16
int	two's complement integer	32
long	two's complement integer	64
float	IEEE 754 floating point	32
double	IEEE 754 floating point	64

Reading input from console

Java program to read inputs from console and find their sum

```
import java.util.Scanner;

public class Addition
{
    public static void main( String[] args )
    {
        Scanner input = new Scanner( System.in );

        int number1;
        int number2;
        int sum;

        System.out.print( "Enter first integer: " );

        number1 = input.nextInt();

        System.out.print( "Enter second integer: " );

        number2 = input.nextInt();

        input.close();

        sum = number1 + number2;
        System.out.println("Sum is: "+sum);

    }
}
```

Importing class Scanner from package util

Package is a bundle of classes

To read from System input (keyboard) using built-in Scanner class of Java

Reading an integer

GPP: We should close the scanner object after its use

What is takeaway from last program?

- ❑ Scanner class is used to read from System input
- ❑ We need to import classes/packages if we need to use them in our program
- ❑ Packages are also known as java class library or Java Application Programming Interface (Java API)
- ❑ We can import a single class from a package
- ❑ Scanner can also be used to scan from a file or a string

```
Scanner sc = new Scanner (new File ("myNumbers") );  
  
String input = "One two three four five six seven";  
  
Scanner s = new Scanner(input).useDelimiter(" ");
```



What is takeaway from last program?

❑ Default package in Java

- ✓ No need to import explicitly
- ✓ e.g. `java.lang` (supports fundamental classes)

❑ Java supports a huge number of packages

- ✓ <https://docs.oracle.com/javase/7/docs/api/overview-summary.html>

❑ We should only import the packages/classes that we use in our program

We should learn how and when to use these packages to solve a given problem!

Conditional statement in Java

Java program to read inputs from console and to compare them

```
import java.util.Scanner;

public class Addition
{
    public static void main( String[] args )
    {
        Scanner input = new Scanner( System.in );

        int number1;
        int number2;

        System.out.print( "Enter first integer: " );

        number1 = input.nextInt();

        System.out.print( "Enter second integer: " );

        number2 = input.nextInt();

        if( number1 < number2)
            System.out.println(number1+" is less than "+number2 );
        else
            System.out.println( number1+" is greater than or equal to "+number2 );
    }
}
```

Loop and switch statement in Java

A simple Java program with loop and switch statement

```
import java.util.Scanner;

public class Addition {
    public static void main( String[] args ) {
        Scanner input = new Scanner( System.in );

        for(int i=0; i<3; i++) {

            int number;
            System.out.print( "Enter the integer: " );
            number = input.nextInt();

            switch(number) {

                case 0:
                    System.out.println("You typed zero");
                    break;

                case 1:
                    System.out.println("You typed one");
                    break;

                default:
                    System.out.println("Neither zero nor one");
            }
        }
    }
}
```

Loop, control statements are similar to C

Array: Introduction

- ❑ Collection of same type of data items (e.g. **int**, **Book**)
- ❑ An array has a size and it remains same throughout its life (**maximum number of elements it can store**)
- ❑ Each element in an array can be accessed by its index (e.g. **c[2]**)

c[0]	24
c[1]	21
.	89
.	56
.	-8
	0
c[6]	55

Array in Java

- ❑ We can create array of primitive types (e.g. **int**, **float**) or reference types (e.g. **String**, **Account**, **Book**)
- ❑ Need to create using **new**

```
int[ ] c = new int[7];  
  
or,      int c[ ];  
         c = new int[7];
```
- ❑ Each element in an array is initialized to 0 (for **primitive** types) or null (for **reference** types)
- ❑ Array can also be declared as
int[] n = { 10, 20, 30, 40, 50 };
- ❑ A lot of supports available (e.g. **n.length** will give 5)

Array in Java

A simple Java program to demonstrate array

```
class ArrayExample {  
  
    public static void main(String args[]) {  
  
        int [ ] set = { 10, 20, 30, 40, 50};  
  
        for(int index=0; index < set.length; index++)  
            System.out.println ( set[index] );  
  
        for(int element : set)  
            System.out.println(element);  
  
        int [ ] set2 = set;  
  
        for(int element : set2)  
            System.out.println(element);  
        }  
}
```

set.length returns size of the array

Iterating through the elements of an array → **enhanced for**

Can we change the values in the array in case of enhanced for? Try!

set2 is a reference variable whose content is same as **set**

❑ We do not need to import any package for the above code

Passing array as arguments

A simple Java program passing array as argument to a method

```
class ArrayExample {  
  
    public static void main(String args[]) {  
  
        int [ ] set = { 10, 20, 30, 40, 50};  
  
        for(int index=0; index < set.length; index++)  
            System.out.print ( set[index] );  
  
        modifyArray(set);  
  
        for(int index=0; index < set.length; index++)  
            System.out.print ( set[index] );  
    }  
  
    public static void modifyArray( int [] auxArray) {  
  
        for(int index=0; index < auxArray.length; index++)  
            auxArray[index] *= 2;  
    }  
  
}
```

\$ 10 20 30 40 50

\$ 20 40 60 80 100

set is passed as parameter in
method modifyArray()

set is a reference variable and thus
changes are reflected in actual
parameter

Work to Do

- ❑ Try to implement all the problems/programs covered in CS101

