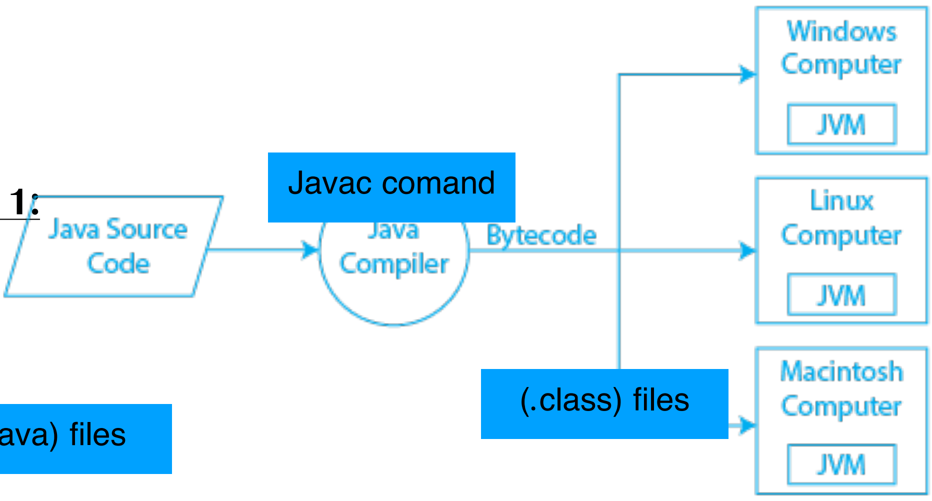
بِسْمِ اللّـهِ الرَّحْمَـٰنِ الرَّحِيـمِ

تلخيص المادة النظرية لمادة جافا ١
Comp 1331

يحتوي التّلخيص على أهم النقاط النظرية التي وردت في المادة، ثمّ بعض الاسئلة المتعلقة بها التي جاءت في الامتحانات السابقة.

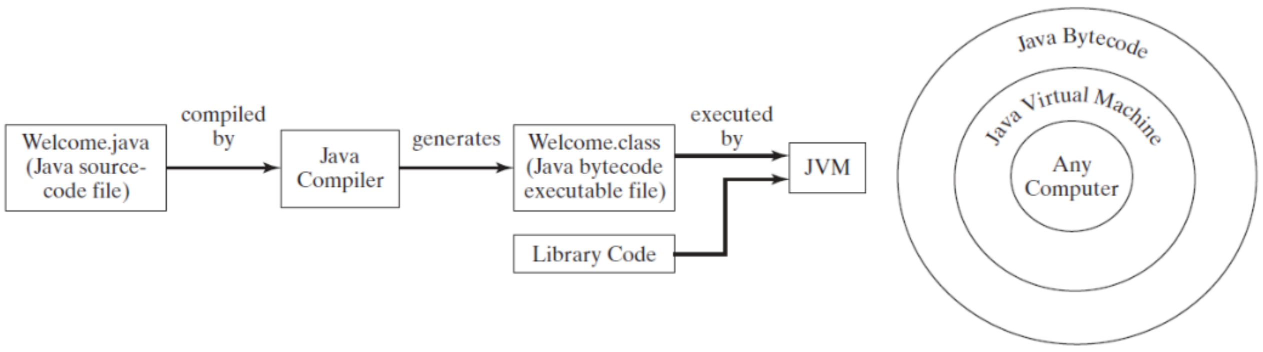
إيمان الغلبان - أصيل قدح

Chapter 1:



Compilation process:

To run the source code of a Java program, we need a Java compiler and a Java interpreter. The Java compiler compiles source files (.java) to generate bytecode (.class) files. We can use the javac command provided by Sun in the terminal window.



(a) (b)

The Java Virtual Machine (JVM) is used to load .class files into the memory and interpret the code. The Execution Engine or Just-In-Time compiler (JIT) (executable name: java) reads one statement from the bytecode in memory, translates it to the machine code of the target platform, and then executes it right away.

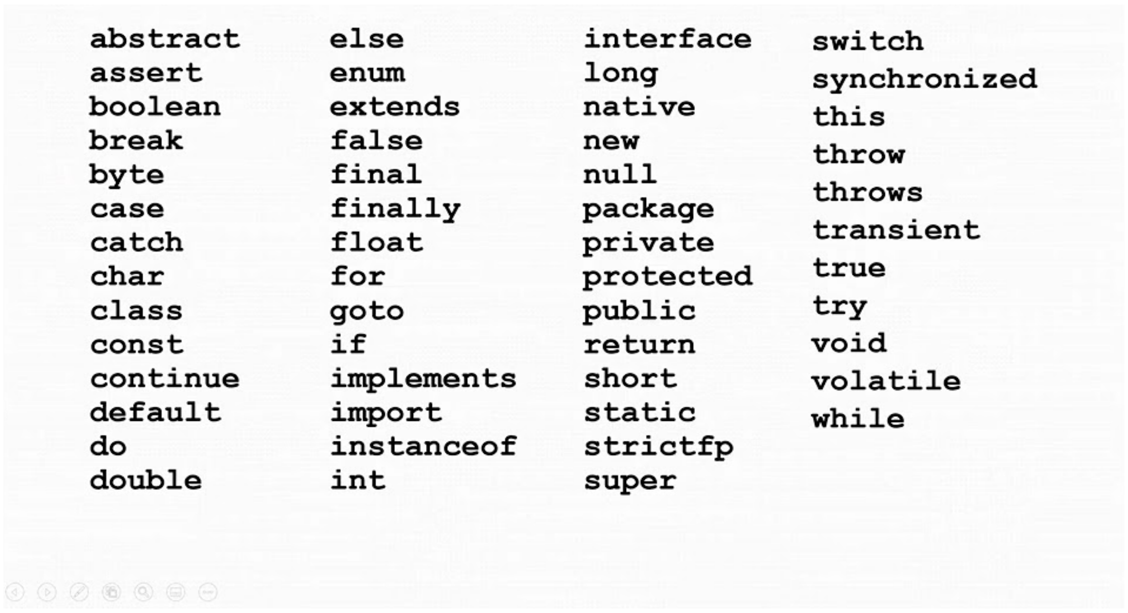
JVM helps to avoid the need to recompile the source code for every specific platform.

Chapter 2:

Reserved words:

Reserved words or keywords are words that have a specific meaning to the compiler and cannot be used for other purposes in the program.

كلمات لها معان محددة يتعرف عليها الـ compiler ولا نستطيع استخدامها لأغراض أخرى.



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

Programming Errors:

- Syntax errors. (Compiling errors)
- Run time errors.
- Logic errors.

Common Errors:

- Undeclared/Uninitialized Variables and.

لا يمكن تعريف متغير دون اعطائه قيمة أولية.

- Round-off Errors. : Calculations involving floating-point numbers are approximated because these numbers are not stored with complete accuracy.

```
System.out.println(1.0 - 0.9);
0.09999999999999998 is displayed
```

Identifiers: (اسماء المتغيرات)

An identifier is a sequence of characters that consist of letters, digits, underscores (_), and dollar signs ($).

An identifier must start with a letter, an underscore (_), or a dollar sign ($). It cannot start with a digit.

An identifier cannot be a reserved word. (See Appendix A, "Java Keywords," for a list of reserved words).

An identifier cannot be true, false, or null.

An identifier can be of any length.

اسماء المتغيرات ممكن أن تحتوي حروف وأرقام و(_) أو ($) ولكن لا يمكن أن يبدأ بـرقم، ولا يمكن أن يكون أحد الكلمات المحجوزة.

To define a constant value we use the reserved word "final".

Chapter 3+5

The difference between break and continue:

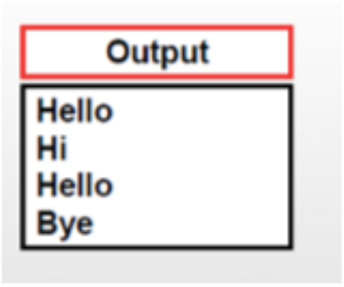
Break	Continue
Keyword ‘break’ is used	Keyword ‘continue’ is used
It is used to terminate the loop	It is used to continue the next iteration in the loop
It is used in switch case and looping statements	It is used in looping statements only.
In break statement, control transfers outside the loop	In continue statement, control remains in the same loop

في حال break بعد تنفيذ الشرط يخرج البرنامج من الـloop،

أما في continue فعندما يصل لها، يكمل لما بعده في الـ loop.

```
public class Example1{
    public static void main (String[]args){
     int i=1;
     while (i++<7){
       System.out.println("Hello");
       if (i==3)
         break;
       System.out.println("Hi");

     }
       System.out.println("Bye");
   }
```


Output

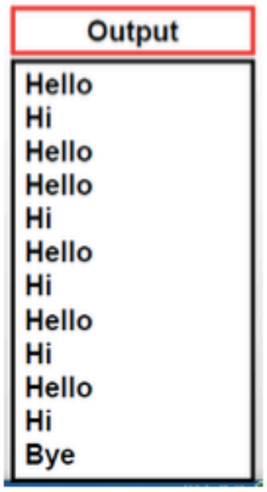
```
Hello
Hi
Hello
Bye
```

```
public class Example2{

    public static void main (String[]args){
     int i=1;
     while (i++<7){
       System.out.println("Hello");
       if (i==3)
         continue;
       System.out.println("Hi");

     }
       System.out.println("Bye");
   }

}
```


Output

```
Hello
Hi
Hello
Hello
Hi
Hello
Hi
Hello
Hi
Hello
Hi
Bye
```

Note:

```
if (i > 0) {
  System.out.println("i is positive");
}
```

(a)

Equivalent

```
if (i > 0)
  System.out.println("i is positive");
```

(b)

في if statement اذا لم يكن هناك اقواس، ففقط أول سطر بعد الـif statemenet يكون تابع لها

```
if (even == true)
   System.out.println(
      "It is even.");
```

(a)

Equivalent

```
if (even)
   System.out.println(
      "It is even.");
```

(b)

Operator	Name	Description
!	not	logical negation
&&	and	logical conjunction
||	or	logical disjunction
^	exclusive or	logical exclusion

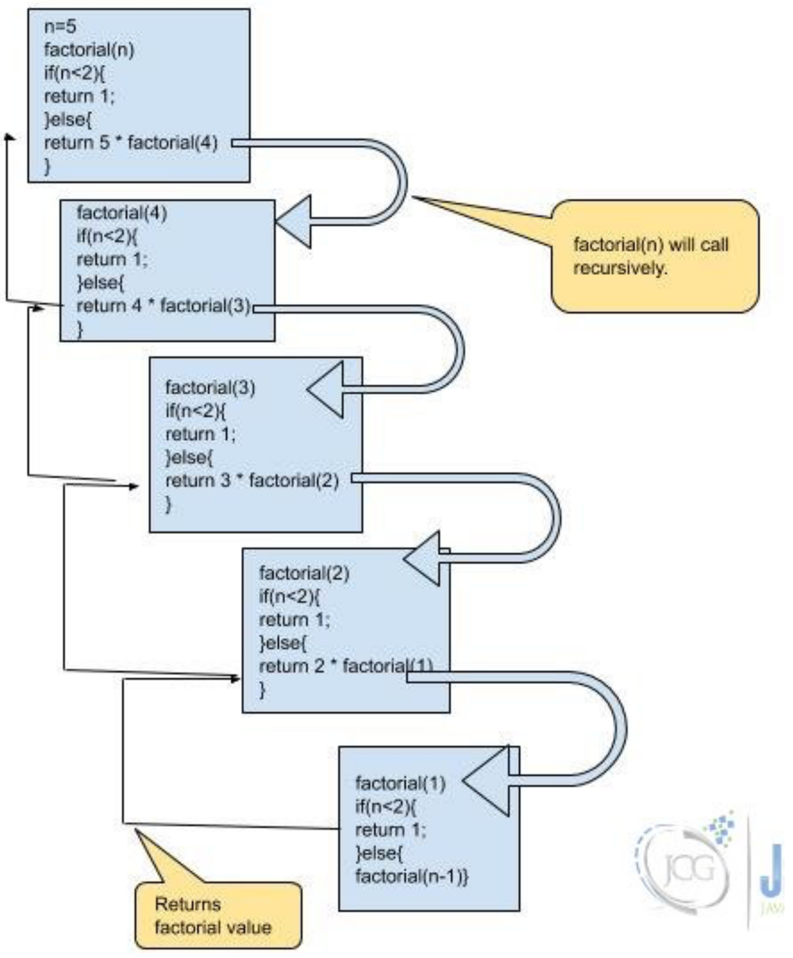
switch Statement Rules:

```
switch (switch-expression) {
  case value1:  statement(s)1;
           break;
  case value2: statement(s)2;
           break;
  default: statement(s)-for-default;
}
```

في جملة الـ switch اذا لم أضع break بعد كلّ case فإن الـ compiler سوف يصل إلى الـ case التي تليها وتطبقها، حتى لو لم يتحقق الشرط.

Recursion:

Recursion is to solve a problem using breaking it into subproblems



Chapter 4:

Rounding Methods:

- **double ceil(double x)**
  x rounded up to its nearest integer. This integer is returned as a double value.

  يتم التقريب لأقرب عدد صحيح أكبر منه، ويتم عرض النتيجة كـdouble.

- **double floor(double x)**
  x is rounded down to its nearest integer. This integer is returned as a double value.

  يتم التقريب لأقرب عدد صحيح أصغر منه، ويتم عرض النتيجة كـdouble.

- **double rint(double x)**
  x is rounded to its nearest integer. If x is equally close to two integers, the even one is returned as a double.

  يتم التقريب لأقرب عدد صحيح، واذا كان بنفس القرب لعددين صحيحين، يتم التقريب لأقرب عدد صحيح زوجي، وتظهر النتيجة كـdouble.

- **int round(float x)**
  Return (int)Math.floor(x+0.5).

  يتم التقريب لأقرب عدد صحيح.

`a + Math.random() * b` $\longrightarrow$ `Returns a random number between a and a + b, excluding a + b.`

Methods in the Character Class:

isDigit (ch):
Returns true if the specified character is a digit

يفحص ان كان الـ chat عبارة عن رقم.

isLetter (ch):
Returns true if the specified character is a letter

يفحص ان كان الـ chat عبارة عن حرف.

isLetterOrDigit (ch):
Returns true if the specified character is a letter or digit

يفحص ان كان الـ chat عبارة عن حرف أو رقم.

isLowerCase (ch):
Returns true if the specified character is a lowercase letter.

يفحص ان كان الـ chat عبارة عن حرف lowercase.

isUpperCase (ch):
Returns true if the specified character is an uppercase letter

يفحص ان كان الـ chat عبارة عن حرف uppercase.

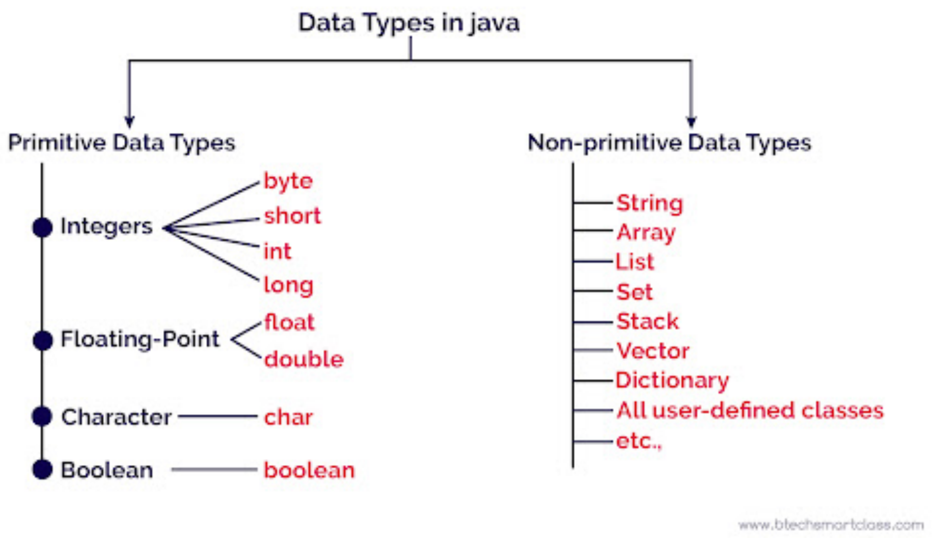
toLowerCase (ch):
Returns the lowercase of the specified character

يعيد الـ lower case.

toUpperCase (ch):
.Returns the uppercase of the specified character

يعيد الـ upper case.

Primitive and non-primitive data types:



The **String** type is not a primitive type. It is known as a reference type. Any Java class can be used as a reference type for a variable. The variable declared by a reference type is known as a reference variable that references an object.

يعتبر الـ string من non primitive type، يسمى الـclass بـ reference ويسمى المتغير المعرف من الـ class بـ reference variable.

Methods in the Character Class:

length ():
Returns the number of characters in this string.
يعيد عدد الـ char الذي يتكون منه الـ string.

charAt (index):
Returns the character at the specified index from this string.
يعيد الـ char الموجود في الموقع المدخل.

concat (s1):
Returns a new string that concatenates this string with string s1.
يعيد string جديد يربط الـ string القديم بـ string جديد مدخل.

toUpperCase():
Returns a new string with all letters in uppercase.
يعيد الـ string نفسه لكن بحروف جميعها uppercase.

toLowerCase():
Returns a new string with all letters in lowercase.
يعيد الـ string نفسه لكن بحروف جميعها lowercase.

trim():
Returns a new string with whitespace characters trimmed on both sides.
يعيد الـ string نفسه لكن بمسافة من الجهتين.

equals (s1):
Returns true if this string is equal to string s1.
يفحص اذا كان الـ strings متساويان (في المحتوى).

equalsIgnoreCase (s1):
Returns true if this string is equal to string s1; it is case insensitive.
يفحص اذا كان الـ strings متساويان (في المحتوى) دون التفريق بين lowerCase أو upperCase.

Methods in the String Class:

compareTo(sl):

Returns an integer greater than 0, equal to 0, or less than 0 to indicate whether this string is greater than, equal to, or less than s1.

يعمل على مقارنة الـ asscii code لـ string ويعيد قيمة حسب الفرق بين الـ asscii code.

compareToIgnoreCase (s1):

Same as compareTo except that the comparison is case insensitive.

يعمل على مقارنة الـ asscii code لـ string ويعيد قيمة حسب الفرق بين الـ asscii code. دون التفريق بين lowerCase أو upperCase.

startsWith (prefix):

Returns true if this string starts with the specified prefix.

يفحص اذا كان الـ string يبدأ بمقطع معين.

endsWith(suffix):

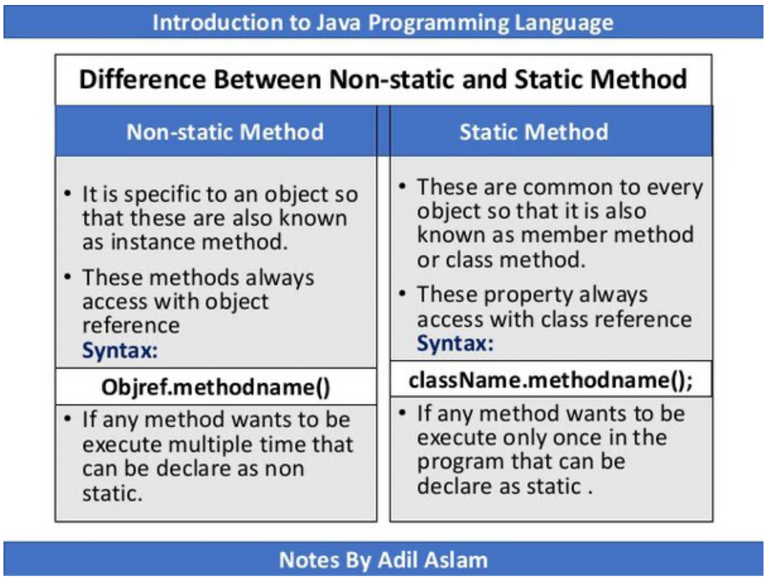
Returns true if this string ends with the specified suffix.

يفحص اذا كان الـ string ينتهي بمقطع معين.

Note:

When **compare two strings (or objects)**, we use the **(==)** operation to compare if they **refer to the same reference or object**, and we use the method **(equals**) to compare **the contents.**

Instance and static Methods


Introduction to Java Programming Language

Difference Between Non-static and Static Method

Non-static Method	Static Method
• It is specific to an object so that these are also known as instance method. • These methods always access with object reference **Syntax:**	• These are common to every object so that it is also known as member method or class method. • These property always access with class reference **Syntax:**
Objref.methodname()	className.methodname();
• If any method wants to be execute multiple time that can be declare as non static.	• If any method wants to be execute only once in the program that can be declare as static .

Notes By Adil Aslam

String methods and Math class methods are both instance, we can't use them except if we use an object.

referenceVariable.methodName(arguments).

A static method can be invoked without using an object.

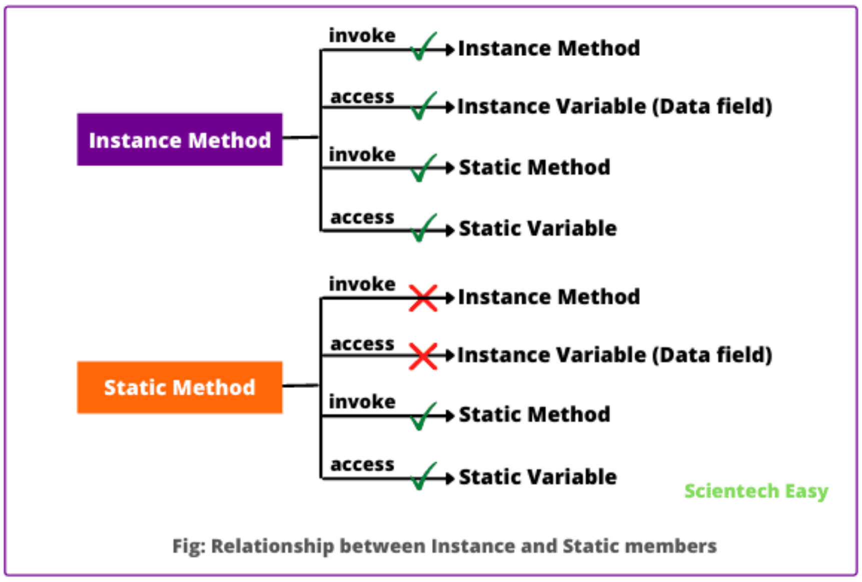
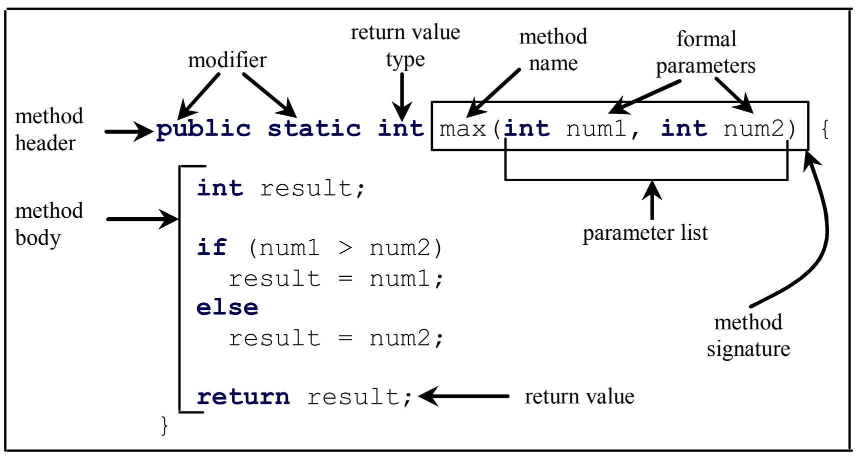


Fig: Relationship between Instance and Static members

static method can't access instance data field or invoke instance methods.

Chapter 6:

Define a method

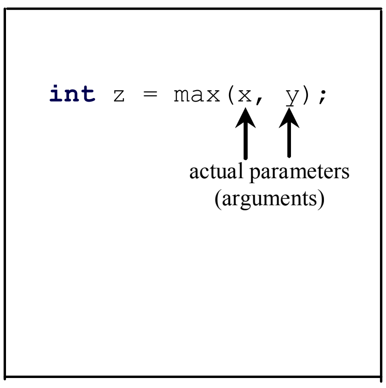


```
public static int max(int num1, int num2) {
  int result;

  if (num1 > num2)
    result = num1;
  else
    result = num2;

  return result;
}
```

Invoke a method



```
int z = max(x, y);
```

Methods in java have two types:

1- returning methods.
2- void methods.

- **A return statement is required for a value-returning method.**
- **Pass by Value means When you invoke a method with an argument, the value of the argument is passed to the parameter.**

Overloading Methods:

Overloading methods enables you to define the methods with the same name as long as their signatures are different.

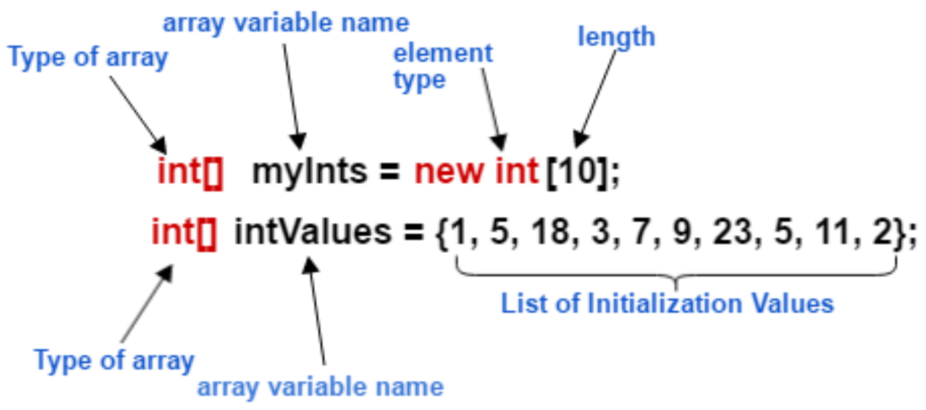
Ambiguous Invocation:

The Java compiler determines which method to use based on the method signature.

If there may be two or more possible matches for an invocation of a method, the compiler cannot determine the most specific match.

This is referred to as ambiguous invocation. Ambiguous invocation is a compile error.

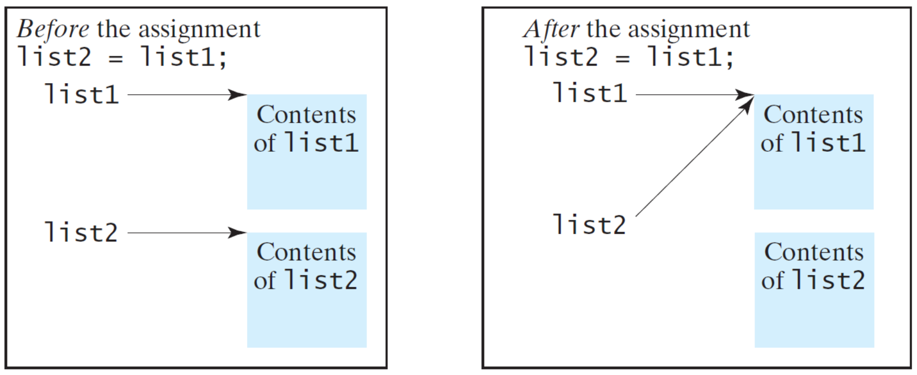
Chapter 7+8 :


array variable name
Type of array
element type
length
int[] myInts = new int[10];
int[] intValues = {1, 5, 18, 3, 7, 9, 23, 5, 11, 2};
List of Initialization Values
Type of array
array variable name

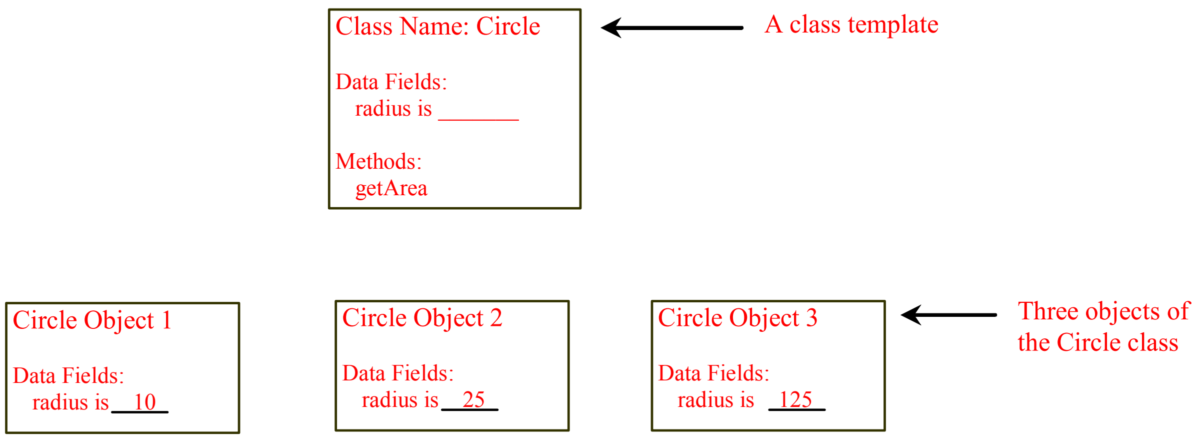
- Java is 0-based language.
- When an array is created, its elements are assigned the default value
- The default values are:

a. 0 for the numeric primitive data types.
b. '\u0000' for char types.
c. false for boolean types.
d. Null for strings.

Copying Arrays:



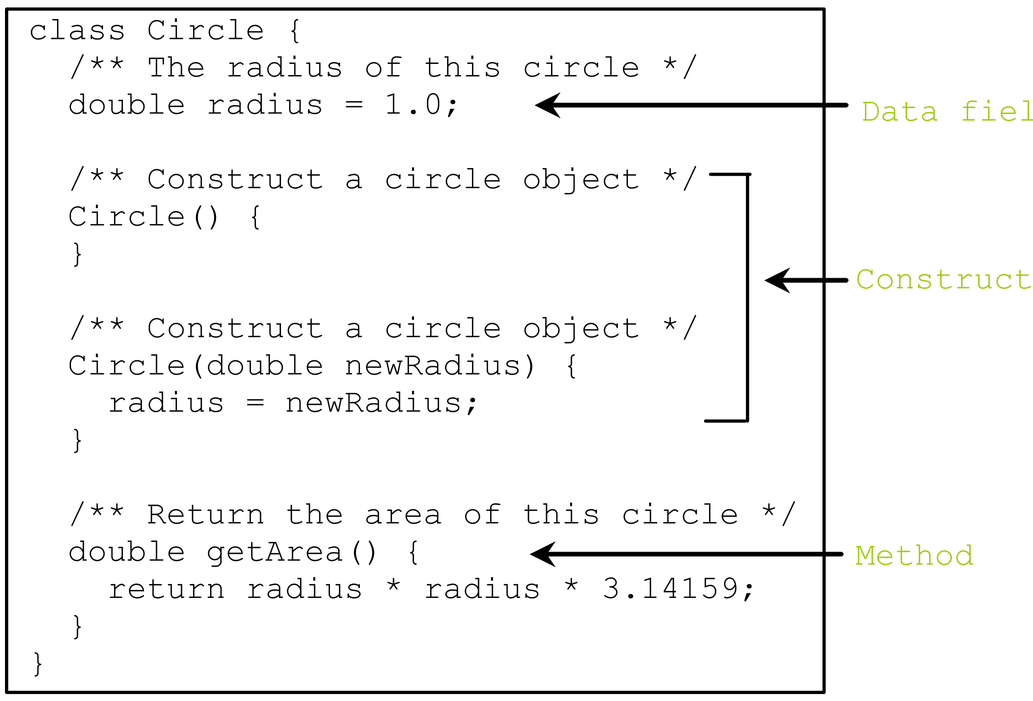
Chapter 9 :



An object has both a state and behavior. The state defines the object, and the behavior defines what the object does.

class provides a special type of methods, known as constructors, which are invoked to construct objects from the class.

Constructors:



```
class Circle {
  /** The radius of this circle */
  double radius = 1.0;

  /** Construct a circle object */
  Circle() {
  }

  /** Construct a circle object */
  Circle(double newRadius) {
    radius = newRadius;
  }

  /** Return the area of this circle */
  double getArea() {
    return radius * radius * 3.14159;
  }
}
```

- A constructor with no parameters is referred to as a no-arg constructor.

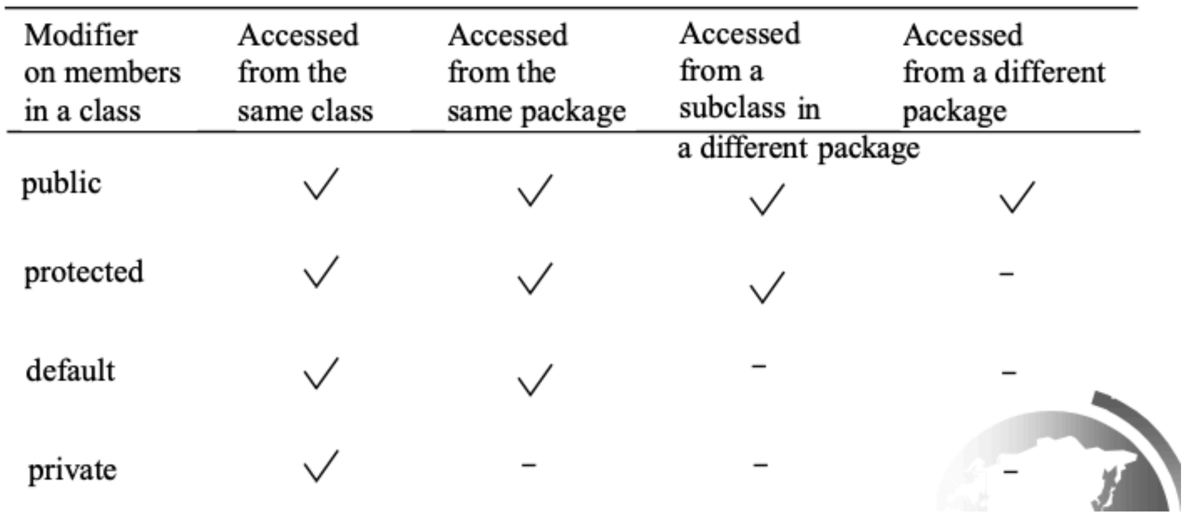
- Constructors must have the same name as the class itself.

- Constructors do not have a return type–not even void.

- Constructors are invoked using the new operator when an object is created.

No - argument constructor is the default constructor, it's provided automatically only if no constructors are explicitly defined in the class

Visibility Modifiers and Accessor/Mutator Methods:



Modifier on members in a class	Accessed from the same class	Accessed from the same package	Accessed from a subclass in a different package	Accessed from a different package
public	✓	✓	✓	✓
protected	✓	✓	✓	–
default	✓	✓	–	–
private	✓	–	–	–

By default, the class, variable, or method can be **accessed by any class in the same package.**

public:

The class, data, or method is visible to any class in any package.

private:

The data or methods can be accessed only by the declaring class.

Public getter (Accessor) and setter (Mutator) methods are used to read and modify private properties (Encapsulation)

Immutable Objects:

If a class is immutable, then all its data fields must be private and it cannot contain public setter methods for any data fields.

A class with all private data fields and no mutators is not necessarily immutable.

The this Keyword :

a. **One common use of the this keyword is reference a class's hidden data fields.**

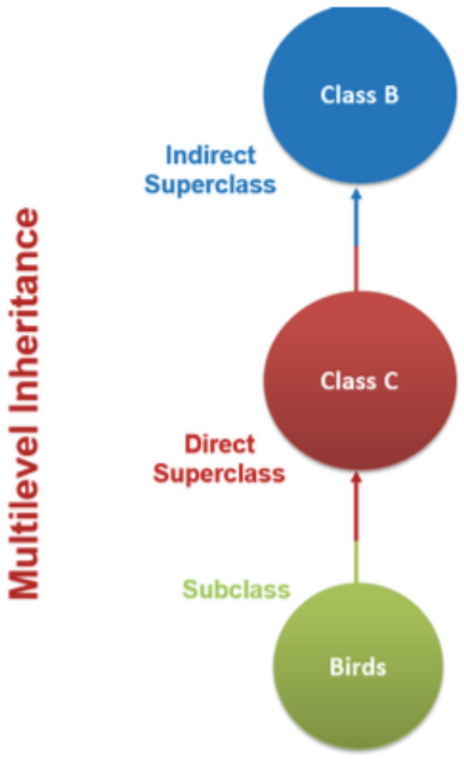
b. **Another common use of the this keyword to enable a constructor to invoke another constructor of the same class.**

Chapter 11:

- a class **C1** extended from another class **C2** is called a subclass, and **C2** is called a superclass.

- A superclass is also referred to as a parent class or a base class,

- subclass as a child class, an extended class, or a derived class



- A subclass inherits accessible data fields and methods from its superclass and

  may also add new data fields and methods.

- Private data fields in a superclass are not accessible outside the class. Therefore, they cannot be used directly in a subclass. They can, however, be accessed/mutated through public accessors and mutators if defined in the superclass.

A subclass and its superclass must have the **is-a** relationship

Superclass's Constructor Is Always Invoked:

A constructor may invoke an overloaded constructor or its superclass's constructor. If none of them is invoked explicitly, the compiler puts super() as the first statement in the constructor.

The super Keyword :

You must use the keyword super to call the superclass constructor. Invoking a superclass constructor's name in a subclass causes a syntax error.

Java requires that the statement that uses the keyword super appear first in the constructor.

The keyword super can also be used to call a method of the superclass as
super.method(parameters).

Overriding	Overloading
Run-time polymorphism	Compile-time polymorphism
Occurs between two classes, using inheritance	Occurs within the class
Array elements are stored in contiguous locations, making it easy to determine relative locations of elements.	Insertion and deletion operations are slow, as elements are supposed to be stored sequentially, index-wise.
Methods involved must have the same name and same signature	Methods involved must have the same name but different signatures

Overriding Methods in the Superclass:

A subclass inherits methods from a superclass. Sometimes it is necessary for the subclass to modify the implementation of a method defined in the superclass. This is referred to as method overriding.

To override a method, the method must be defined in the subclass using the same signature and the same return type as in its superclass.

An instance method can be overridden only if it is accessible. Thus a private method cannot be overridden, because it is not accessible outside its own class. If a method defined in a subclass is private in its superclass, the two methods are completely unrelated.

Like an instance method, a static method can be inherited. However, a static method cannot be overridden. If a static method defined in the superclass is redefined in a subclass, the method defined in the superclass is hidden.

The Object Class and Its Methods:

toString():
returns a string representation of the object. The default implementation returns a string consisting of a class name of which the object is an instance, the at sign (@), and a number representing this object.

equals():
If the equals() method is intended to test whether two objects have the same contents, the method must be overridden in the defining class of the objects.

instanceof:

Use the instanceof operator to test whether an object is an

instance of a class

NOTE:

The == comparison operator is used for comparing two primitive data type values or for determining whether two objects have the same references.

Polymorphism:
objects has “many forms”.

Polymorphism occurs when a program invokes a method through a superclass variable.

At execution time, the correct subclass version of the method is called, based on the type of the reference stored in the superclass variable.

Casting Objects:

Casting from Superclass to Subclass (Downcasting):

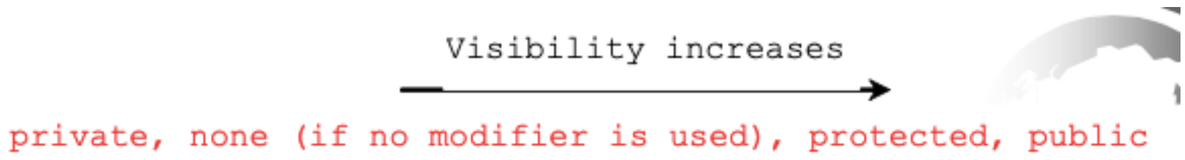
Explicit casting must be used when casting an object from superclass to a subclass.

if the superclass object is not an instance of the subclass, a runtime ClassCastException occurs

NOTE:

A subclass may override a protected method in its superclass and change its visibility to public.

A subclass cannot weaken the accessibility of a method defined in the superclass. For example, if a method is defined as public in the superclass, it must be defined as public in the subclass.



The final Modifier:

You may occasionally want to prevent classes from being extended. In such cases, use the **final** modifier to indicate that a class is final and cannot be a parent class. The **Math** class is a final class.

أسئلة سنوات سابقة للمادة النظرية:

1) **When is downcasting (Explicit) necessary?**

a. To access methods specific to the subclass.
b. Down is not necessary.
c. To make a reference variable more generic.
d. To avoid a ClassCastException.

2) **Choose the correct statements:**

a. The instance function can be called by an object from the class only.
b.The instance function can be called by the name of the class only.
c.The static function can be called by the name of the class only.
d. The static function can be called by an object of the class only.

3) **Which of the following statements is wrong?**

b. Overriding and overloading have the same mean.

c. Overloading a function means the new one has the same name and the same parameters, but you can redefine the code.

d. Override a function means the new one has the same name and the same parameters, but you can redefine the code.

4) **How can a subclass access a protected member of its superclass within a different package?**

a. It cannot be accessed.
b. By casting the superclass reference to the subclass type.
c. By using the super keyword.
d. By using the this keyword

5) Given the following code, which members are accessible from within a different class in the same package?

```
class MyClass {
               public int x;
               private String y;
               protected double z;
           }
```

a. x, y, and z
b. x and z
c. x only
d. z only

6) One of the following statements is False

a. toString() method returns a string representation of the object.
b.toString() method can be hidden in subclass.
c. Every object has a toString method.
d. toString() method can be overridded.

7) Which Statements are true:

a. In case of overloading, binding of objects with methods happens at runtime.

b. In the case of Overriding, the creation of objects happen at compile time and these objects are used for calling objects at runtime

c. In the case of Dynamic polymorphism, method behavior is decided at runtime.

d. In the case of Overriding, it is the responsibility of the compiler to bind the method calls with the method body.

8) Which access modifier is most commonly used for instance variables to promote encapsulation?

a. Private. (in other words, using setter/getter methods)
b. Public.
c. Default.
d. Protected.

9) Which Statements are true:

a. The compiler dynamically binds the implementation of the method at runtime, decided by the actual class of the object referenced by the variable.

b. None of the above.

c. The JVM finds a matching method according to parameter type, number of parameters, and order of the parameters at compile time.

d. A method may be implemented in several subclasses

10) What is the toString() method used for?

- To provide a string representation of an object.

11) What is the purpose of the final modifier?

- To prevent class extending and method overriding.

12) What does a subclass inherit from its superclass?

- Accessible data fields and methods

13) What is the difference between instance and static variables and methods?

- **Instance variables and methods belong to individual objects, while static variables and methods belong to the class itself.**

14) Which of the following is generated when the source code is successfully compiled?

a. Output.
b. Source code.
c. Byte code.
d. None of above.

15) The JDK command to compile a class in the file Test.java is:

a. Java Test.
b. Javac Test .Java
c. Java Test.
d. Java Test.class

16) which of the following is not java reserved word?

a. Null.
b. Import.
c. Args.
d. Package.

16) In java, if you do not give a value to an int local variable before using it:

a. Compiler will give an error
b. It will initialized to zero
c. It will contain a garbage value.
d. None of above.

17) The following are all java primitive types, except:

a. String
b. Boolean
c. Byte
d. Char

18) to Declare a constant in java, we must use:

a. Static
b. Final
c. Void
d. Double

19) which statement is true about default constructor?

a. Compiler creates always a default constructor if it's not written.
b. Compiler creates a default cinstructor if only there are no other constructors.
c. Compiler is not creating a default constructor at all.
d. None of the above.

20) When must a program explicitly use the "this" reference?

a. Accessing a private variable.
b. Accessing a public variable.
c. Accessing an instance variable with the same name (shadowed) by local variable.
d. Accessing a local variable.

21) the overloaded method is?

a. Method with the same name and same signature.
b. Method with the different name and different signature.
c. Method with the same name and different signature..
d. Method with the different name and same signature.

22) which statement is true regarding instance and static?

a. Static methods can access both instance and static members.
b. instance methods can access both instance and static members.
c. Static methods can access only instance members.
d. instance methods can access only static members.

22) which statement is true regarding instance and static?

a. Static methods can access both instance and static members.
b. instance methods can access both instance and static members.
c. Static methods can access only instance members.
d. instance methods can access only static members.

23) which statement is true ?

a. A reference variable is an object.
b. A reference variable references to an object
c. A data field in a class must be of a primitive type.
d. A data field in a class must be of a object type.

24) encapsulation means ?

a. That data fields should be declared private.
b. That a class can extend another class.
c. That a variable of supertype can refer to subtype object.
d. That a class can contain another class.

25) when invoking a method with an object argument, ________ is passed.

a. The contents of the object
b. A copy of the object
c. The reference of the object
d. The object is copied, then the reference of the copied object

27) which of the following is not true about an immutable object?

a. All the contents of an immutable object cannot be modified.
b. All properties of an immutable object must be private.
c. All properties of an immutable object must be primitive type.
d. an immutable object contains no mutator methods.

28) particular member belongs to a type itself rather than to an instance of the type and shared across all instance of the class.

a. Paublic.
b. private.
c. Static.
d. class.

29) What is the primary purpose of a class in Java?

A) To create methods only
B) To allocate memory
C) To define a new data type by grouping variables and methods
D) To execute code

30) Which keyword is used to create a new instance of a class?

A) class
B) new
C) this
D) instance

31) What is the default constructor of a class?

A) A constructor that takes no arguments
B) A constructor provided by the Java compiler if no constructors are explicitly defined
C) A constructor that initializes all member variables.
D) The first constructor in a class.

32) Which of the following is true about a static method?

A) It can only be called on an instance of a class
B) It can access instance variables directly
C) It belongs to the class, rather than any object of the class
D) It must return a value

33) How is a constant defined in a Java class?

A) Using the const keyword
B) Using the final keyword
C) By initializing a variable at the time of declaration
D) By declaring it static

34) What is encapsulation in Java?

A) The process of wrapping code and data together into a single unit
B) A design pattern
C) The ability of an object to take many forms
D) The concept of inheriting properties from a class

35) Which keyword is used to inherit a class in Java?

A) super
B) extends
C) implements
D) this

36) How do you call a superclass constructor from a subclass in Java?

A) Using super() in the subclass constructor
B) Using this() in the subclass constructor
C) By directly calling the superclass constructor method
D) It is called automatically by Java runtime

37) What does it mean to override a method in Java?

A) To change the method's return type in a subclass
B) To provide a new implementation for a method in the subclass
C) To remove a method from the superclass in the subclass
D) To call a method from the superclass in the subclass

38) Which type of error will be in the following program:
System.out.println(9.0-8.9); ?

A) Logical error.
B) Round-off error.
C) Run time error.
D) None of the above.

39) Which of the following is NOT a valid variable name?

A) radius2
B) 2radius
C) area2circle
D) radius

40) What does JVM stand for?

A)Java Visual Model.
B) JavaScript Virtual Machine
C)Java Virtual Machine.
D) none.

40 *What is polymorphism in Java?
A) The ability of Java to execute methods faster.
B) The capability of a class to extend multiple classes.
C) The concept of allowing methods to do different things based on the object that it is acting upon.
D) The method of hiding data within classes.

41) How does Java implement dynamic polymorphism?
A) Through static methods
B) By method overloading
C) By method overriding
D) Through constructors

42) Which of the following is not a type of inheritance in Java?
A) Single Inheritance
B) Multiple Inheritance
C) Multilevel Inheritance
D) Hierarchical Inheritance

43) What does the keyword final signify when applied to a method?
A) The method can be overridden
B) The method cannot be overridden
C) The method can be overloaded
D) The method will return a final value

تــــمَّ بِـــحَــــــمــــــدِ الـــلّه.

نسأل الله التّوفيق لنا ولكم..