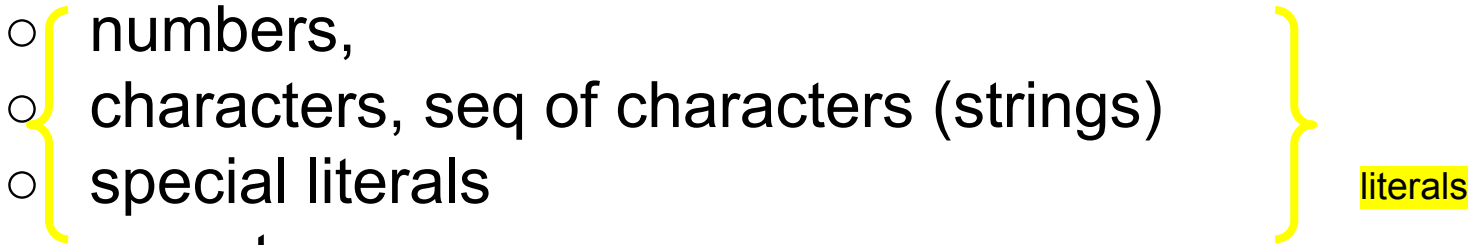


Tinker Academy

AP Computer Science Prep (Java DS & Algo)
Lecture 2 - Java Fundamentals 2
(Keywords, Identifiers)

Lecture 2 - Java Fundamentals 2

Compilation Process

- Java source code split into “token” sequence
- Tokens are either
 - numbers,
 - characters, seq of characters (strings)
 - special literals

literals

- operators
- language keywords
- identifiers
- punctuation, comments, whitespace

Language Keywords

Lecture 2 - Java Fundamentals 2

Language Keywords

- Keywords are reserved
- You cannot create identifiers with the same name as the keyword
- Keywords are case sensitive (as is everything else in Java)

Lecture 2 - Java Fundamentals 2

Keywords (Reserved words)

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

Lecture 2 - Java Fundamentals 2

Java Program

```
/**
 * HelloWorld is a Java Class
 * The entire code for HelloWorld is contained in this file "HelloWorld.java"
 */
public class HelloWorld {
    // main is a Java Method. Other programs can send messages to the main method
    public static void main(String[] args) {
        String message = "Hello World";
        System.out.println(message);
    }
}
```

HelloWorld.java

Identifiers

Lecture 2 - Java Fundamentals 2

Identifiers

- Name of a class
- Name of a method
- Name of a variable
- Name of a package
- Name of an interface

Lecture 2 - Java Fundamentals 2

Java Program

```
/**
 * HelloWorld is a Java Class
 * The entire code for HelloWorld is contained in this file "HelloWorld.java"
 */
public class HelloWorld {
    // main is a Java Method. Other programs can send messages to the main method
    public static void main(String[] args) {
        String message = "Hello World";
        System.out.println(message);
    }
}
```

HelloWorld.java

Punctuation

Lecture 2 - Java Fundamentals 2

Punctuation

- Curly Braces {} around Class Body
- Curly Braces {} around Method Body
- Semicolon ; after a Java Statement
- Parenthesis () around method parameters
- Parenthesis () around method invocation
- Dot between parts of a package name
- Dot between name of a package and class
- Dot between name of class and name of method
- Square brackets [] around array types

Lecture 2 - Java Fundamentals 2

Java Program

```
/**
 * HelloWorld is a Java Class
 * The entire code for HelloWorld is contained in this file "HelloWorld.java"
 */
public class HelloWorld {

    // main is a Java Method. Other programs can send messages to the main method
    public static void main ( String[] args ) {

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

    }

}
```

HelloWorld.java

Comments

Lecture 2 - Java Fundamentals 2

Comments

- Multi line Comments
 - across multiple lines
 - Starts with `/*` (forward slash and star)
 - Ends with `*/` (star and forward slash)
- Single line Comments
 - on a single line
 - Starts with `//`
- Compiler will ignore comments, comments are to help understand the code

Lecture 2 - Java Fundamentals 2

Java Program

```
/**  
 * HelloWorld is a Java Class  
 * The entire code for HelloWorld is contained in this file "HelloWorld.java"  
 */  
public class HelloWorld {  
  
    // main is a Java Method. Other programs can send messages to the main method  
    public static void main ( String[] args ) {  
  
        System.out.println( "Hello World" );  
  
    }  
  
}
```

HelloWorld.java

Whitespace

Lecture 2 - Java Fundamentals 2

Whitespace

- Anything that is orange is a whitespace
 - Single space
 - Newline
 - Tab
- Newline and tab characters are not displayed on most editors. You can view them in Eclipse. Refer to the Getting Started with Eclipse guide

Lecture 2 - Java Fundamentals 2

Java Program

```
/**
 * HelloWorld is a Java Class
 * The entire code for HelloWorld is contained in this file "HelloWorld.java"
 */
public class HelloWorld {

    // main is a Java Method. Other programs can send messages to the main method
    public static void main ( String[] args ) {

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

    }

}
```

HelloWorld.java

Class Activity