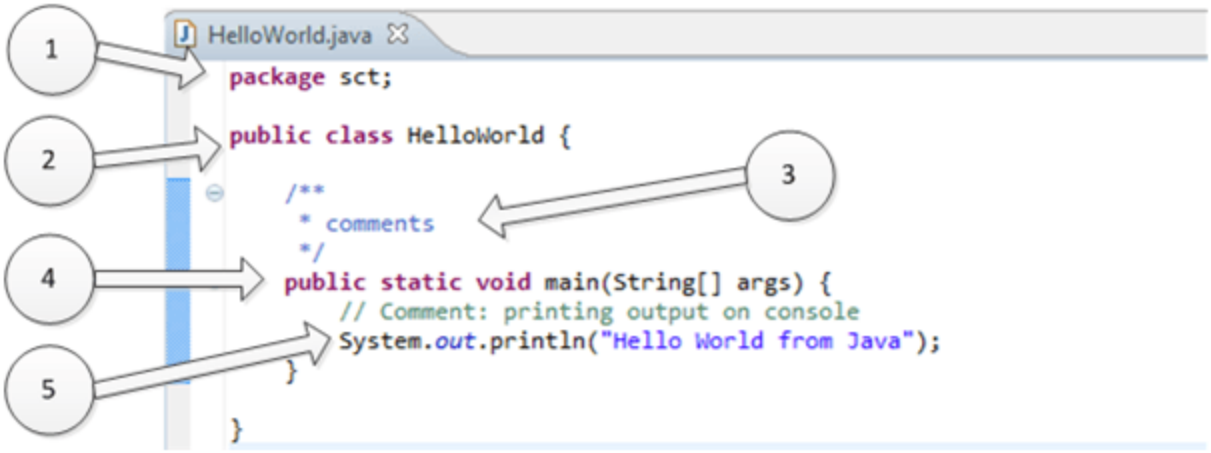
Anatomy of a Java program:

Let's use the example of HelloWorld Java program to understand structure and features of the class. This program is written on few lines, and its only task is to print "Hello World from Java" on the screen. Refer to the following picture:



package sct:
It is package declaration statement. The package statement defines a namespace in which classes are stored. The package is used to organize the classes based on functionality. If you omit the package statement, the class names are put into the default package.

Public class HelloWorld:
public: This is access modifier keyword which tells compiler access to class. Various values of access modifiers can be public, protected, private or default (no value).
class: This keyword used to declare a class. Name of class (HelloWorld) followed by this keyword.

Comments section:
In java, we can write comments in the following two ways:
a. Line comments: It starts with two forward slashes (//) and continues to the end of the current line. Line comments do not require an ending symbol.
b. Block comments start with a forward slash and an asterisk (/*) and end with an asterisk and a forward slash (*/). Block comments can also extend across as many lines as needed.

public static void main (String [ ] args):
Its method (Function) named **main** with string array as an argument.
a. public: Access Modifier
b. static: static is a reserved keyword which means that a method is accessible and usable even though no objects of the class exist.
c. void: This keyword declares nothing would be returned from the method. The method can return any primitive or object.
d. Method content inside curly braces. { }

System.out.println("Hello World from Java"):
a. System: It is the name of Java utility class.
b. out: It is an object which belongs to System class.
c. println: It is a utility method name which is used to send any String to the console.
d. "Hello World from Java": It is String literal set as argument to println method.

Common Programming guidelines:

- Java identifiers must start with a letter, a currency character ($), or a connecting character such as the underscore (_). Identifiers cannot start with a number. After first character identifiers can contain any combination of letters, currency characters, connecting characters, or numbers. For example:
    - int variable1 = 10; //This is valid
    - int 4var =10; // This is invalid, identifier can't start with a digit.
- Identifiers, method names, class names are case-sensitive; var and Var are two different identifiers.
- You can't use Java keywords as identifiers.

List of java keywords:

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

Primitive data type:

Not everything in Java is an object. There is a special group of data types (also known as primitive types) that will be used quite often in programming. For performance reasons, the designers of the Java language decided to include these primitive types. Java determines the size of each primitive type. These sizes do not change from one operating system to another. This is one of the key features of the language that makes Java so portable. Java defines eight primitive types of data: byte, short, int, long, char, float, double, and boolean. The primitive types are also commonly referred to as simple types which can be put in four groups:

- **Integers:** This group includes byte, short, int, and long, which are for whole-valued signed numbers.
- **Floating-point numbers:** This group includes float and double, which represent numbers with fractional precision.
- **Characters:** This group includes char, which represents symbols in a character set, like letters and numbers.
- **Boolean:** This group includes boolean, which is a special type for representing true/false values.