Programming Fundamentals

Fall 2021

Lecture 2
Dr. Farhan Aadil
farhan.aadil@ciit-attock.edu.pk
DrAadil.com

Please turn OFF your Mobile Phones!

Today’s Lecture

- Programming in a nutshell
- Java Vs C++
- Integrated Development Environment - IDE
- Programing Basics

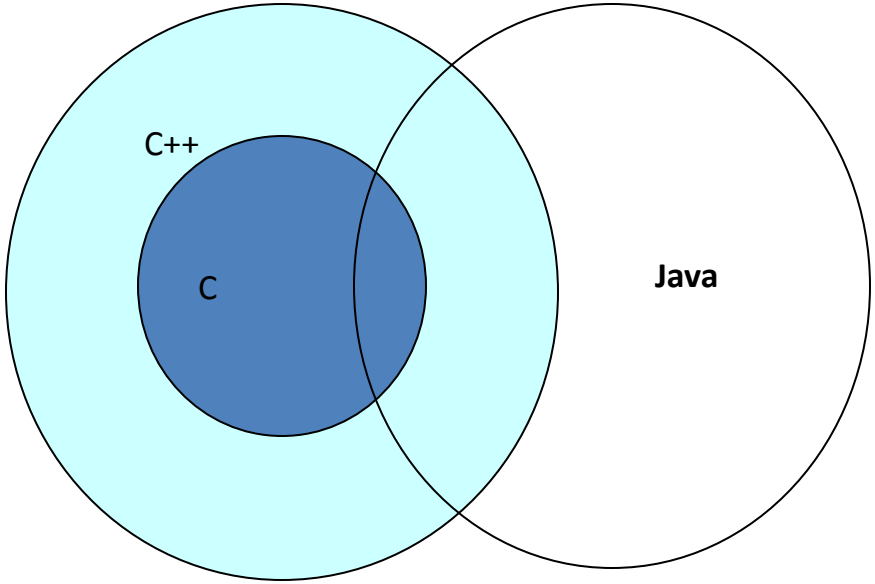
What is a Computer Program?

- For a computer to be able to do anything (multiply, play a song, run a word processor), it must be given the instructions to do so.
- A **program** is a set of instructions written by humans for computers to perform tasks.
- The instructions are written in **programming languages** such as C, C++, Java, etc.

Java

Year	Development
1990	Sun decided to developed special software that could be used for electronic devices. A project called Green Project created and head by James Gosling.
1991	Explored possibility of using C++, with some updates announced a new language named “Oak”. The name Oak was used by Gosling after an oak tree that stood outside his office.
1992	The team demonstrated the application of their new language to control a list of home appliances using a handheld device.
1993	The World Wide Web appeared on the Internet and transformed the text-based interface to a graphical rich environment. The team developed Web applets (time programs) that could run on all types of computers connected to the Internet.
1995	Oak was renamed to Java

Overlap of C, C++, and Java

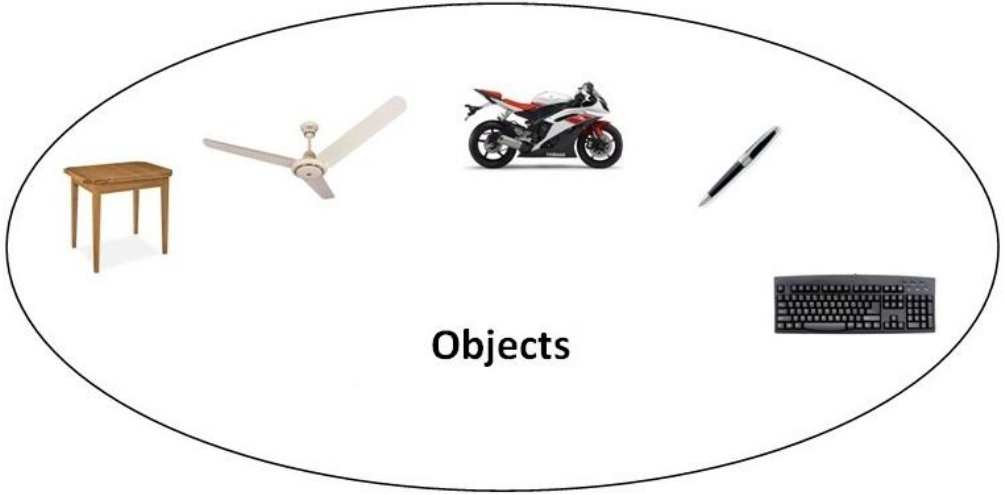


Java better than C++ ?

- No Pointers
- No Unsafe Structures
- No Multiple Inheritance

Object-Oriented Programming

An **object** is anything that can be represented by data in a computer's memory and manipulated by a computer program.



Object-Oriented Programming

An object can be something in the physical world or even just an abstract idea.

An airplane, for example, is a physical object that can be manipulated by a computer.

Integrated Development Environment (IDE)

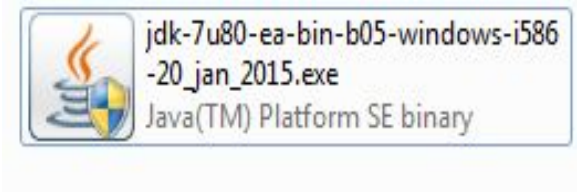
Lecture 2

What is Eclipse?

- Best known as an Integrated Development Environment (IDE)
  - Provides tools for coding, building, running and debugging applications
- Eclipse started as an IBM product
- Open Source
  - It is a general-purpose open platform that facilitates and encourages the development of third-party plug-ins
- Originally designed for Java, now supports many other languages
  - Good support for C, C++
  - Python, PHP, Ruby, etc…

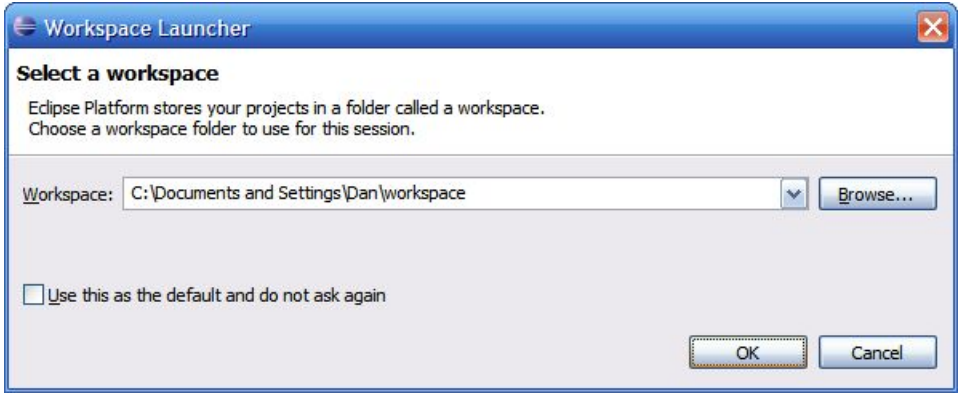
Prerequisites for Running Eclipse

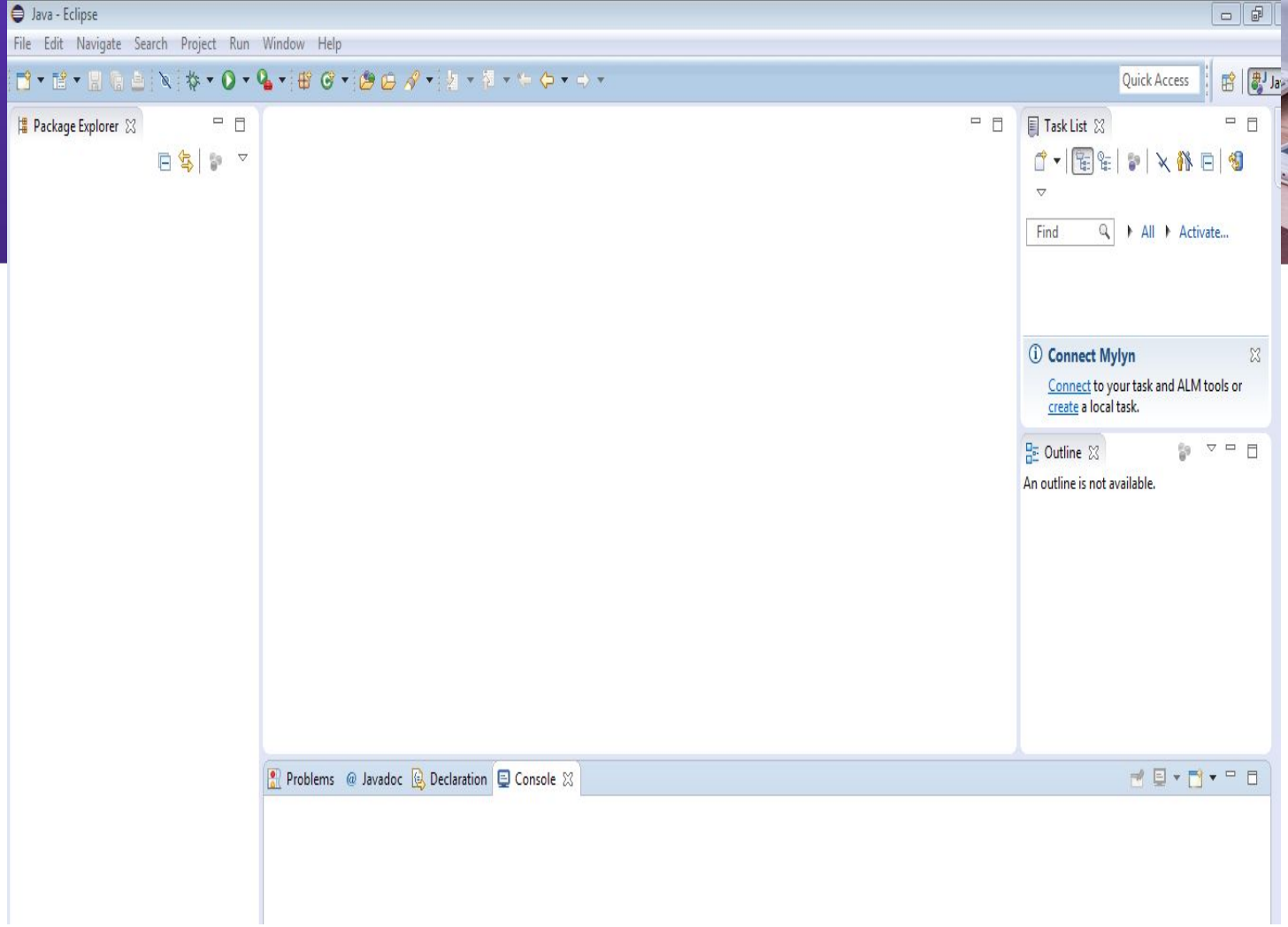
- Eclipse is written in Java and will thus need an installed JDK in which to execute



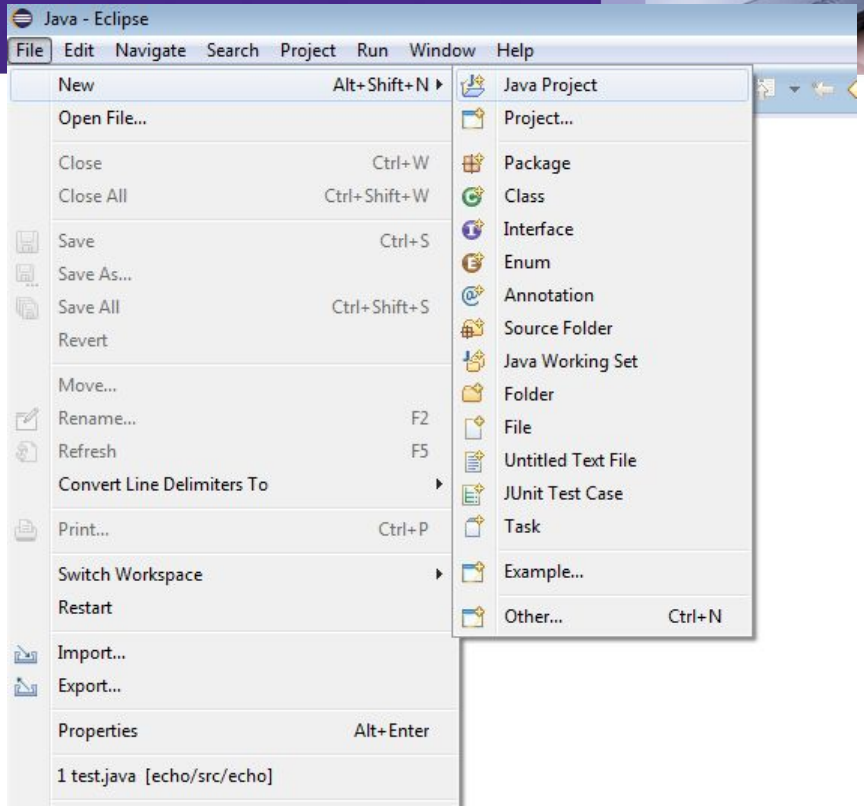
Selecting a Workspace

- In Eclipse, all of your code will live under a workspace
- A workspace is nothing more than a location where we will store our source code and where Eclipse will write out our preferences
- Choose a location where you want to store your files, then click OK

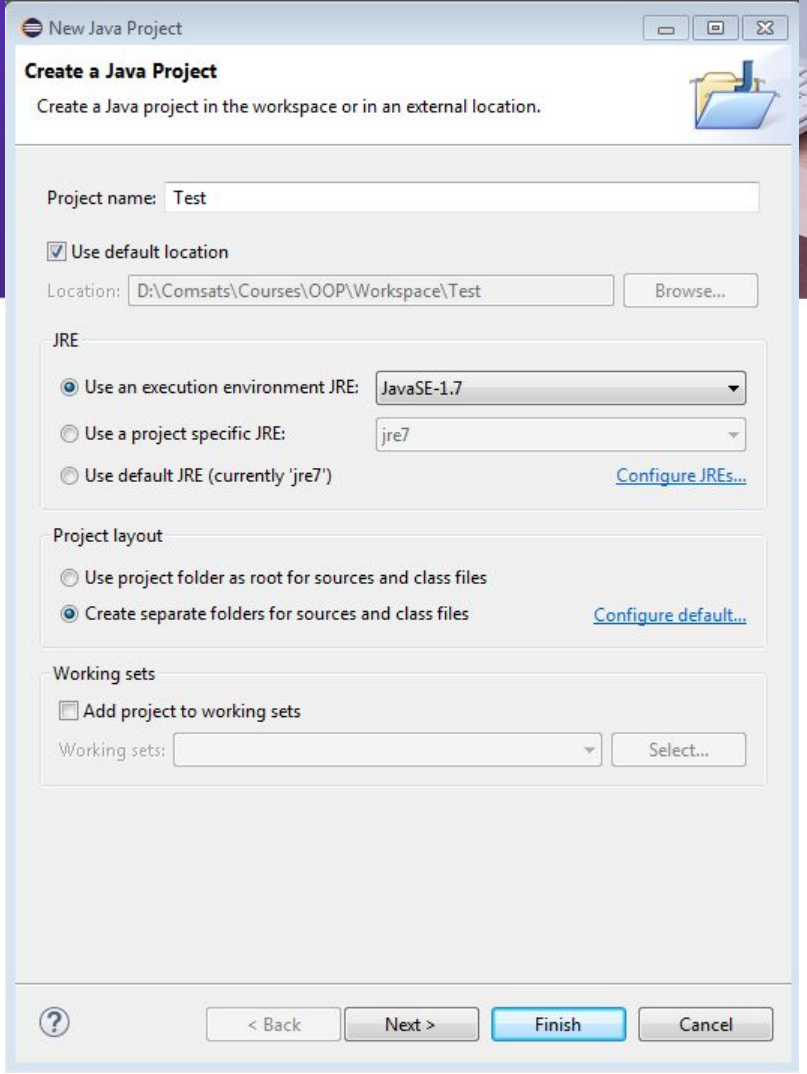



Java - Eclipse
File Edit Navigate Search Project Run Window Help
Quick Access
Package Explorer
Task List
Find
All
Activate...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
An outline is not available.
Problems
Javadoc
Declaration
Console

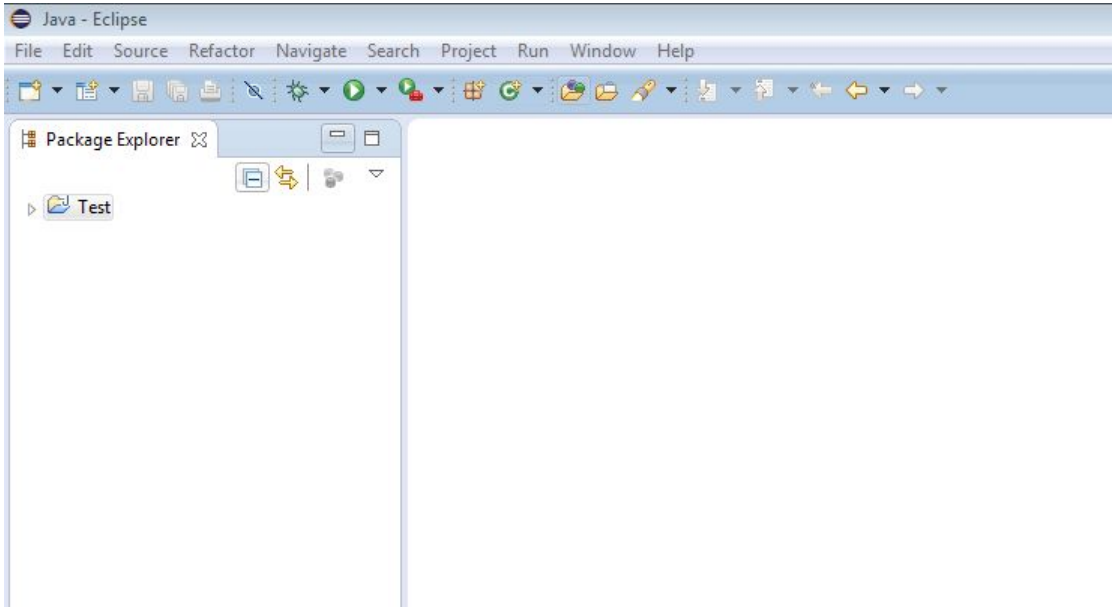
Eclipse Environment



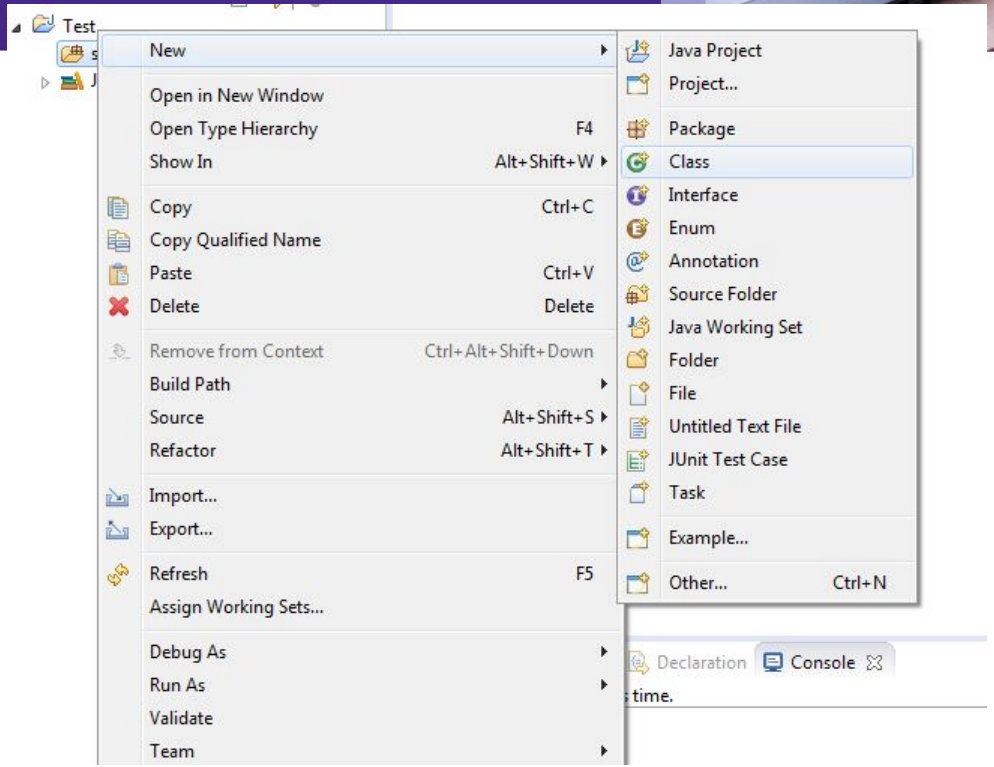
Eclipse Environment



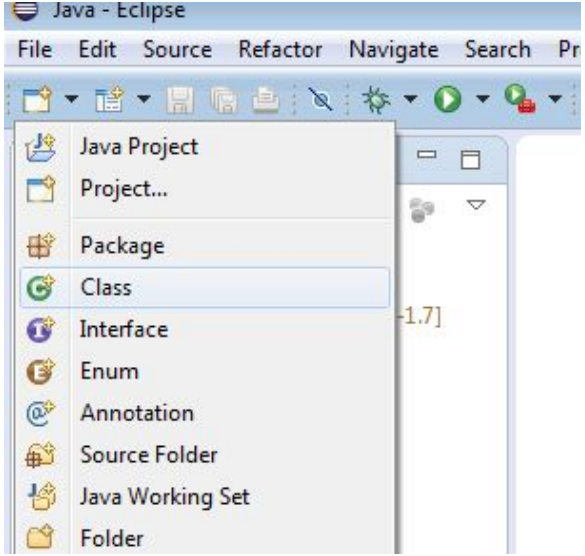
Eclipse Environment



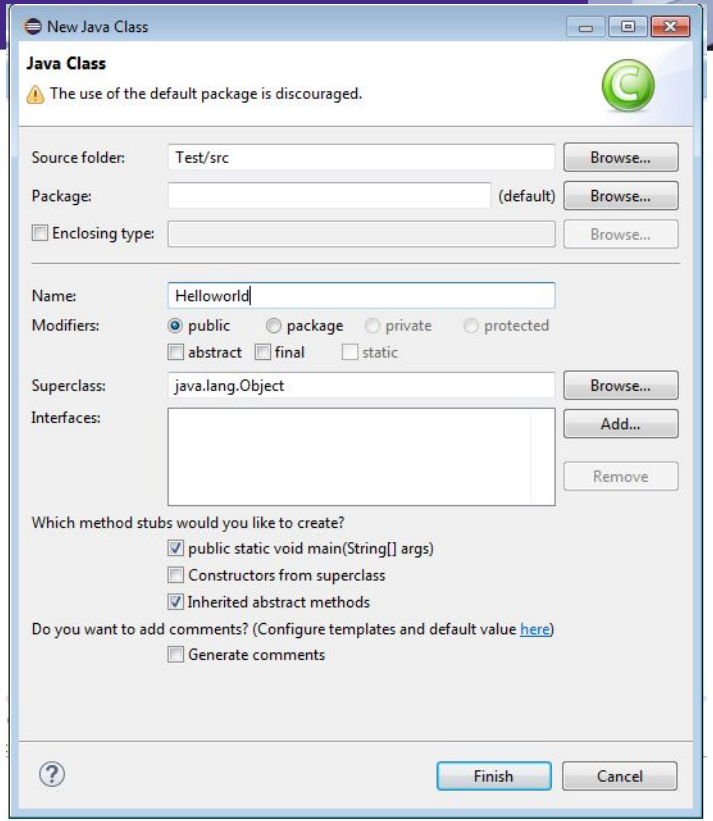
Eclipse Environment



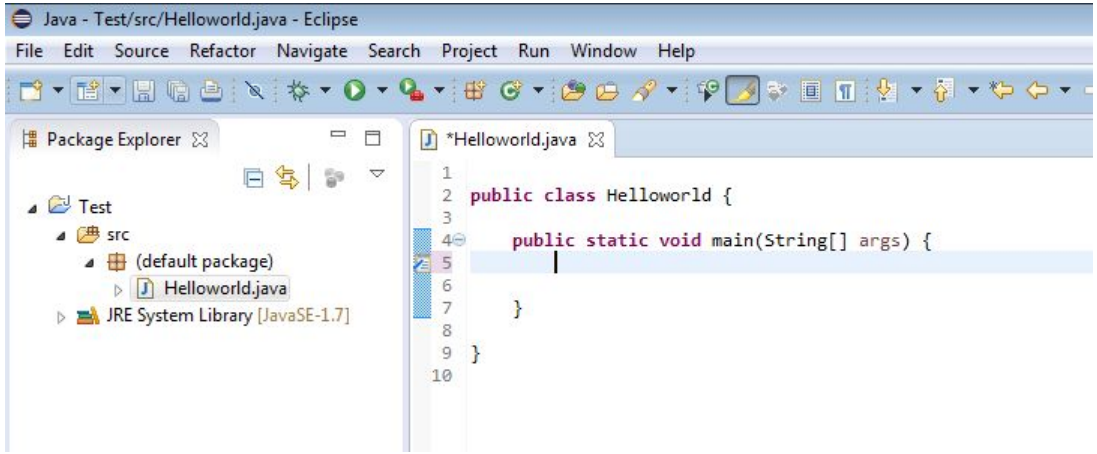
Eclipse Environment



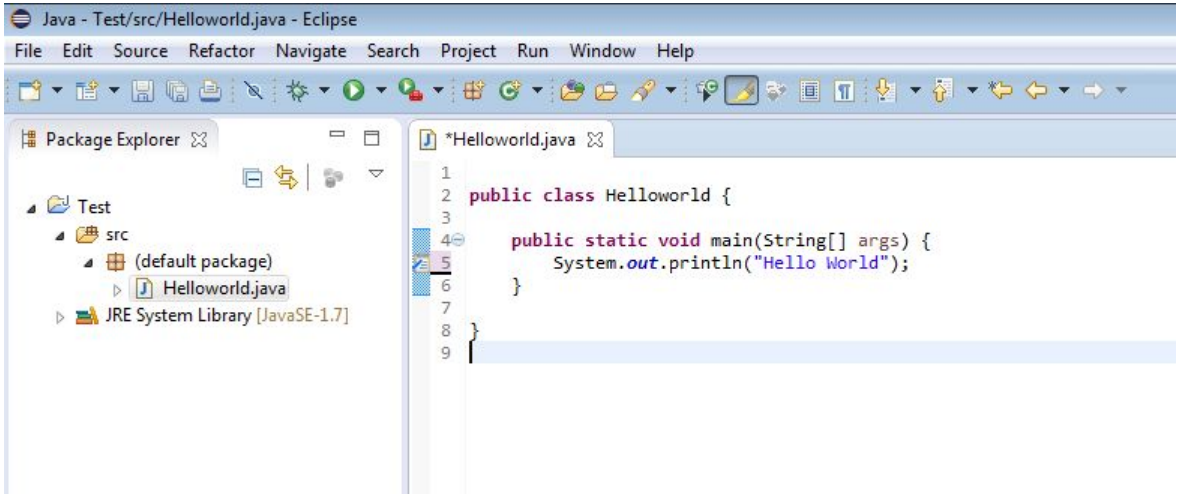
Eclipse Environment

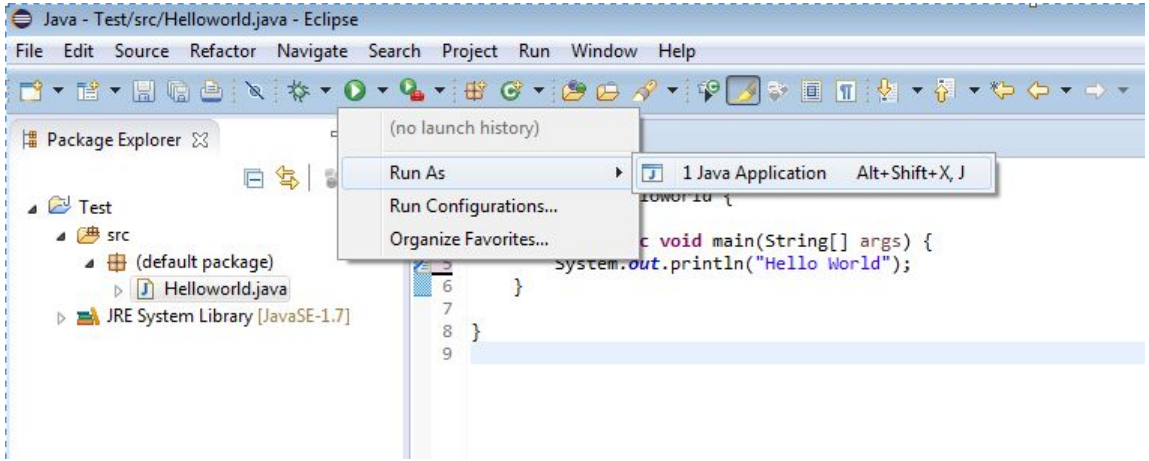


HelloWorld Program



HelloWorld Program




Java - Test/src/Helloworld.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
Test
src
(default package)
Helloworld.java
JRE System Library [JavaSE-1.7]
(no launch history)
Run As
Run Configurations...
Organize Favorites...
1 Java Application Alt+Shift+X, J
c void main(String[] args) {
System.out.println("Hello World");
}
}

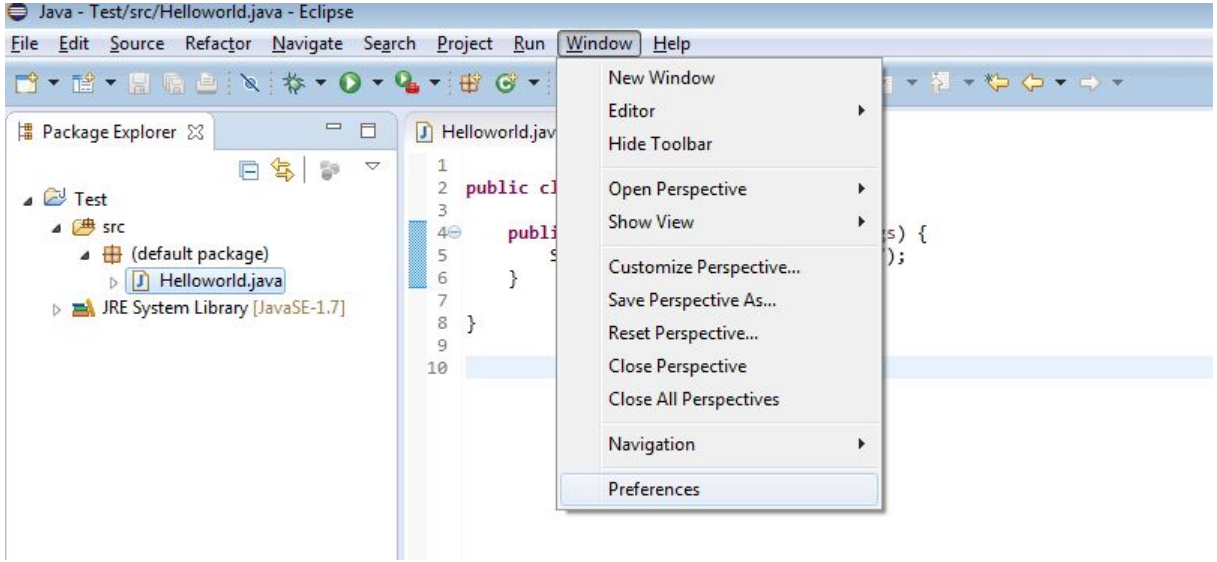
Helloworld.java

```java

public class Helloworld {

    public static void main(String[] args) {
        System.out.println("Hello World");
    }

}

```

Problems @ Javadoc Declaration Console

```
<terminated> Helloworld [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (Feb 9, 2017, 1:30:46 AM)
Hello World
```

HelloWorld Program



Font Size

Preferences

type filter text

- General
  - Appearance
    - Colors and Fonts
    - Label Decorations
  - Compare/Patch
  - Content Types
  - Editors
  - Globalization
  - Keys
  - Network Connectio
  - Perspectives
  - Search
  - Security
  - Startup and Shutdo
  - Web Browser
  - Workspace
- Ant
- Code Recommenders
- Help
- Install/Update
- Java
- Maven
- Mylyn
- Run/Debug
- Team
  - Validation
- WindowBuilder
- XML

Colors and Fonts

Colors and Fonts (? = any character, * = any string):

type filter text

- Text Editor Block Selection Font
- Text Font
- CVS
- Debug
- Git

Edit...

Use System Font

Reset

Font

Font: Consolas

Consolas
Constantia
Corbel
Courier
Courier New
Fixedsys

Font style: Regular

Regular
Italic
Bold
Bold Italic

Size: 12

12
14
16
18
20
22
24

Sample

AaBbYyZz

Script: Western

Show more fonts

OK

Cancel

Outline

Java - Test/src/Helloworld.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access | Java

Package Explorer

- Test

Helloworld.java

```
1  import static java.lang.System.*;
2
3  public class Helloworld {
4
5      public static void main(String[] args) {
6          int a=5;
7          int b= 10;
8          out.println("The sum of a and b "+(a+b));
9          out.print("The sum of a and b "+(a+b));
10
11     }
12
13 }
14
15
```

Outline

- Helloworld
  - main(String[]) : void

Problems | Javadoc | Declaration

0 items

Description	Resource	Path	Location	Type

Writable | Overwrite | 5 : 28

11:29 PM
2/13/2017

Warning

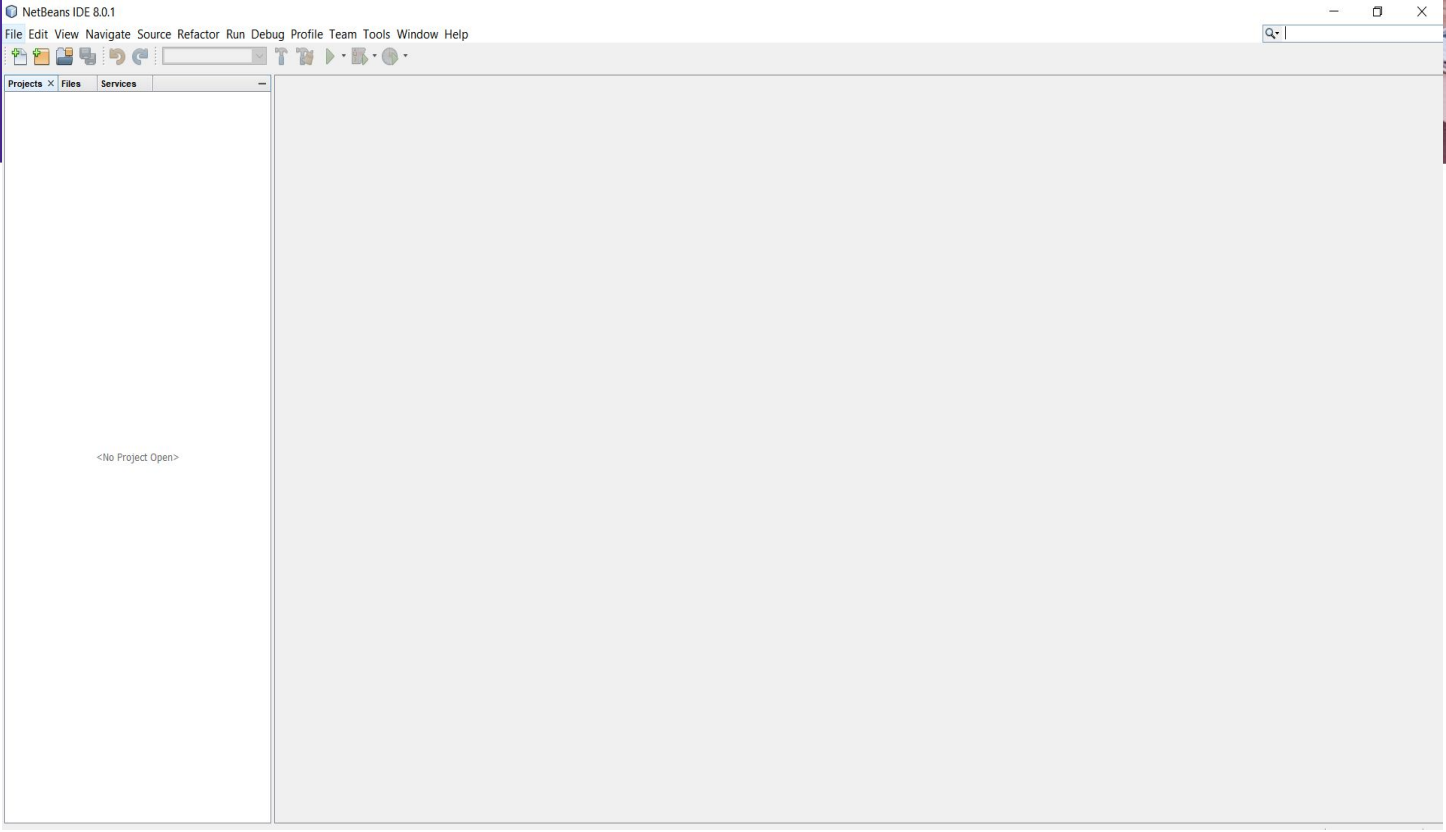
Helloworld.java

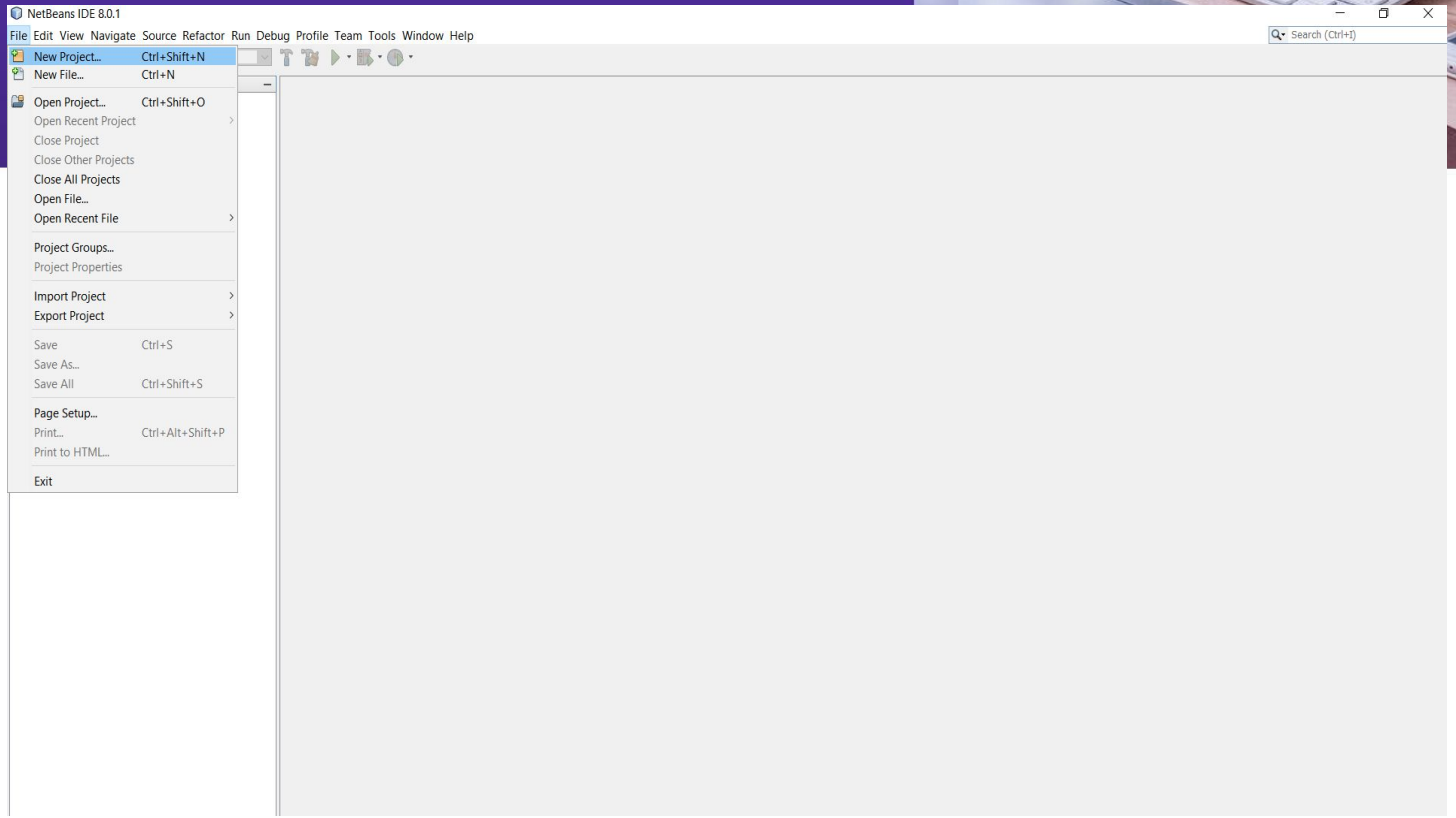
```
1
2  public class Helloworld {
3
4      public static void main(String[] args) {
5          int a=2;
6          int b=3;
7
8
9      }
10
11 }
12
13
```

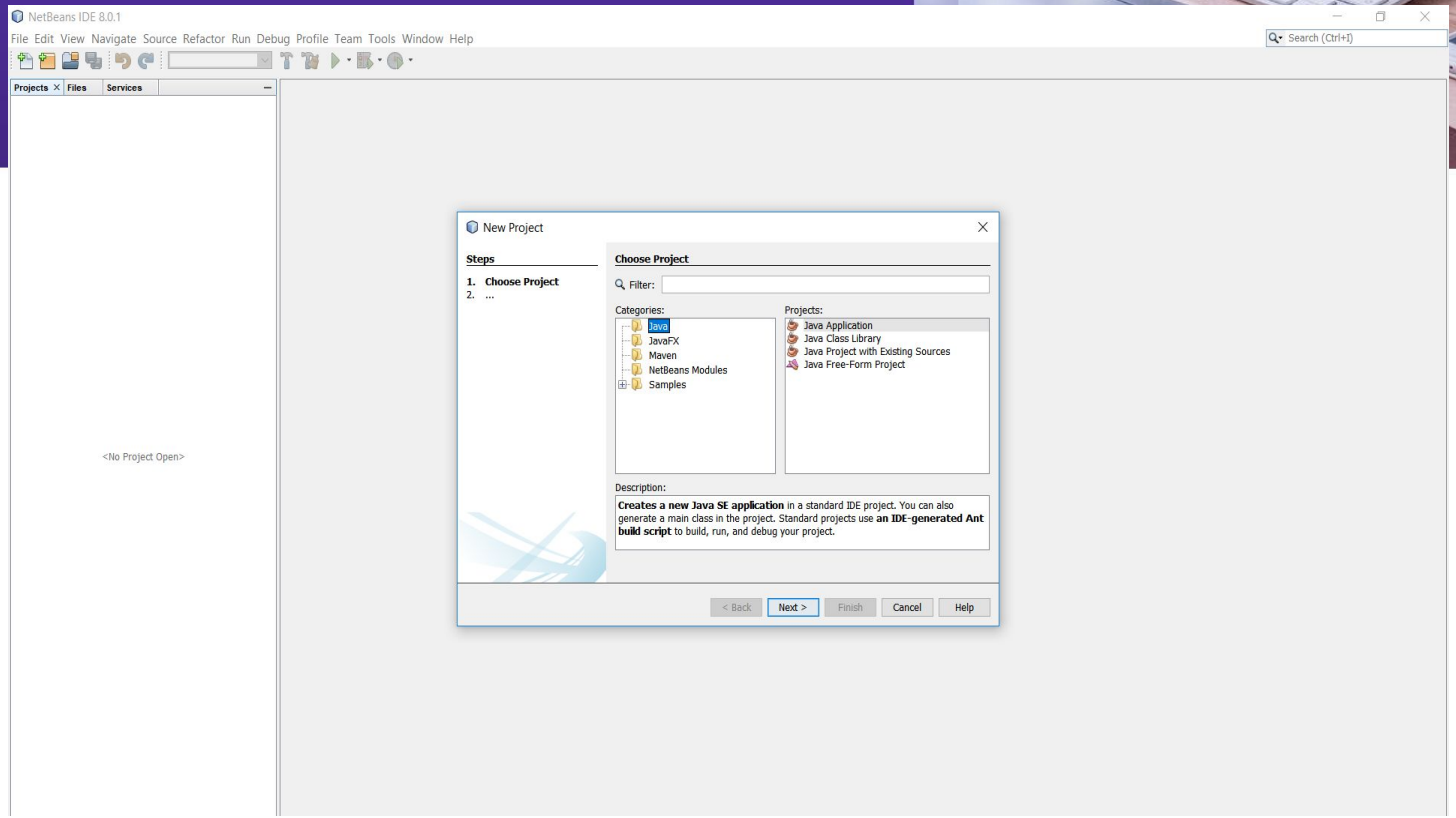
Problems | @ Javadoc | Declaration | Console

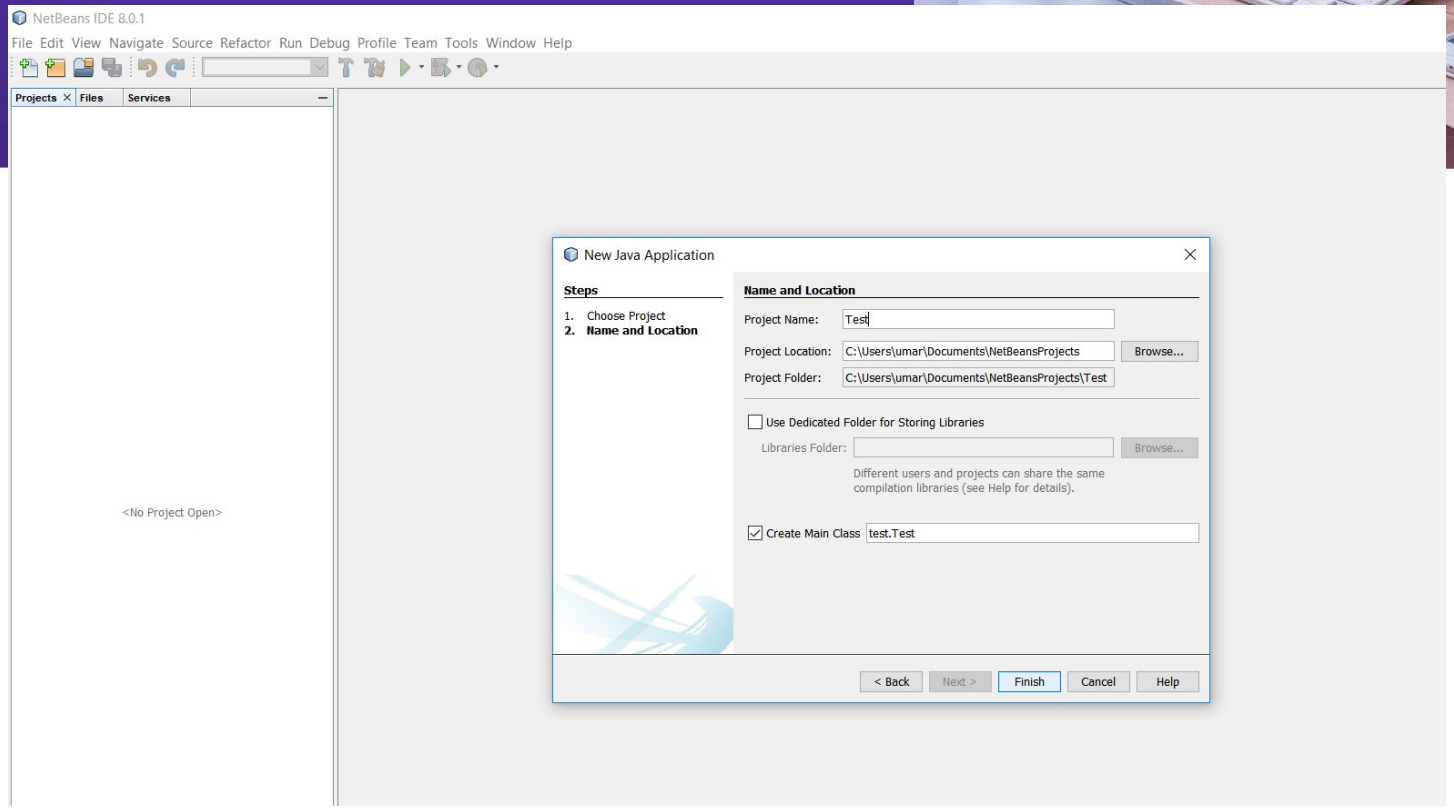
Description	Resource	Path	Location	Type
Warnings (2 items)				
The value of the local variable a is not used	Helloworld.java	/Test/src	line 5	Java Problem
The value of the local variable b is not used	Helloworld.java	/Test/src	line 6	Java Problem

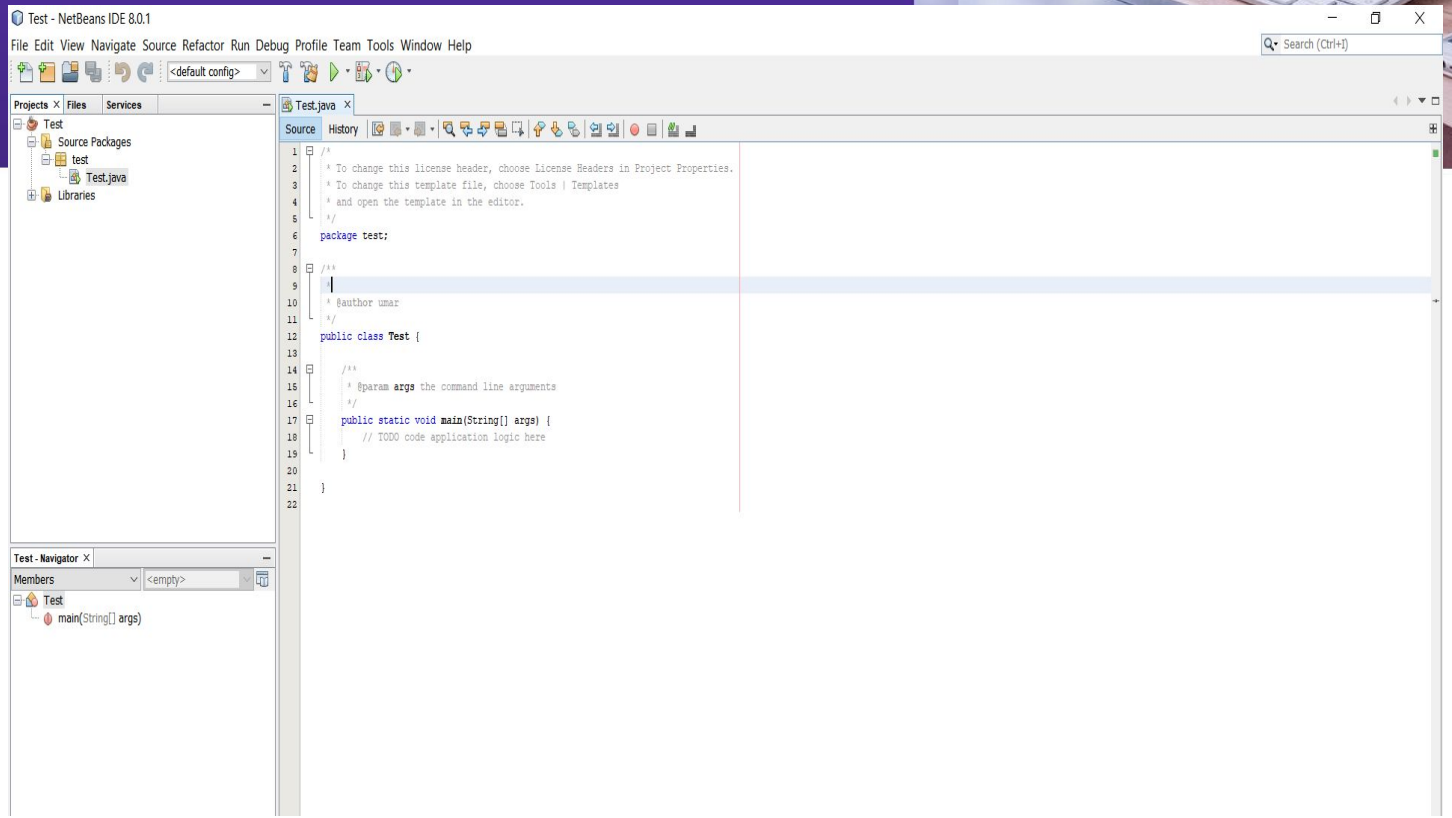
NetBeans

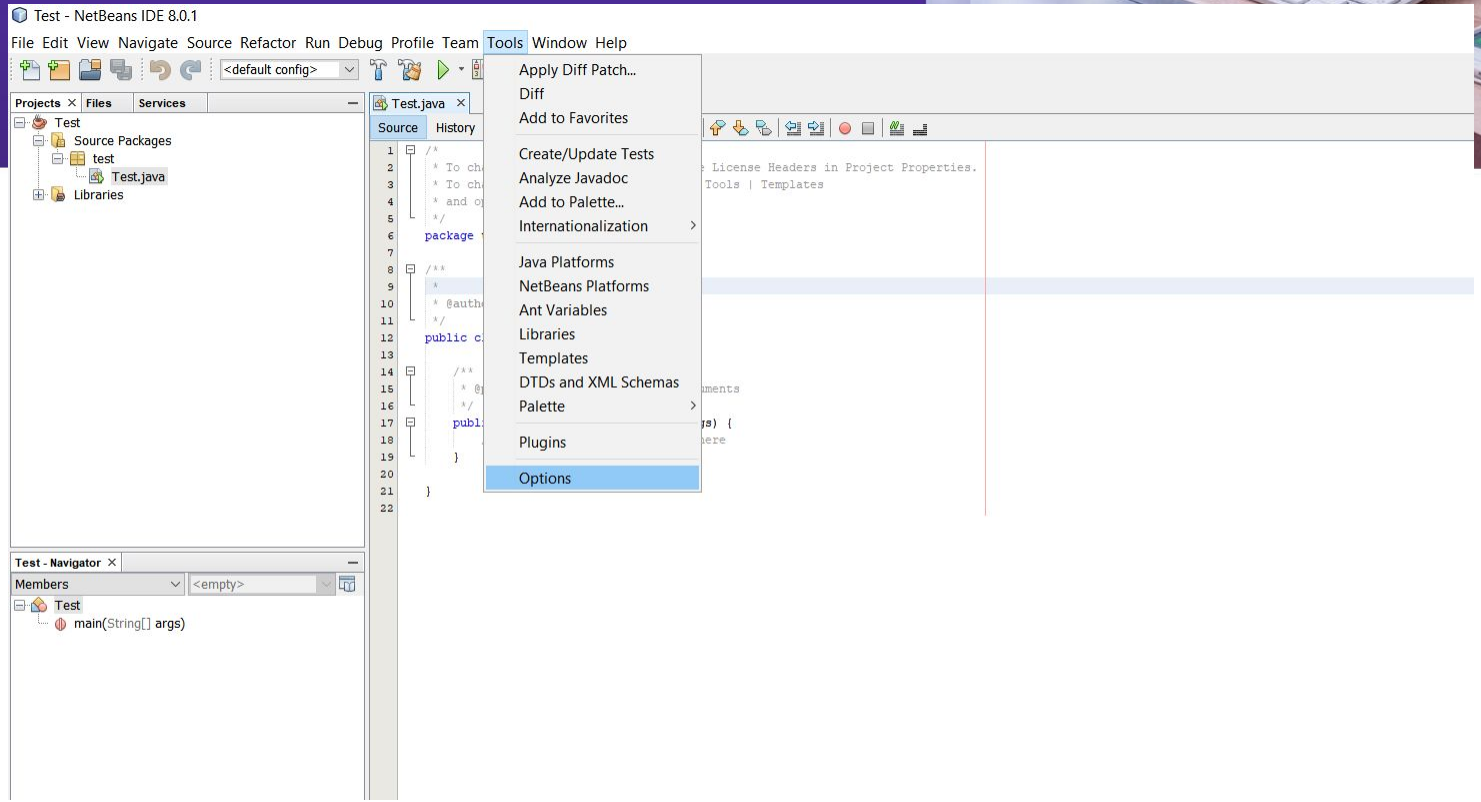

NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
<No Project Open>

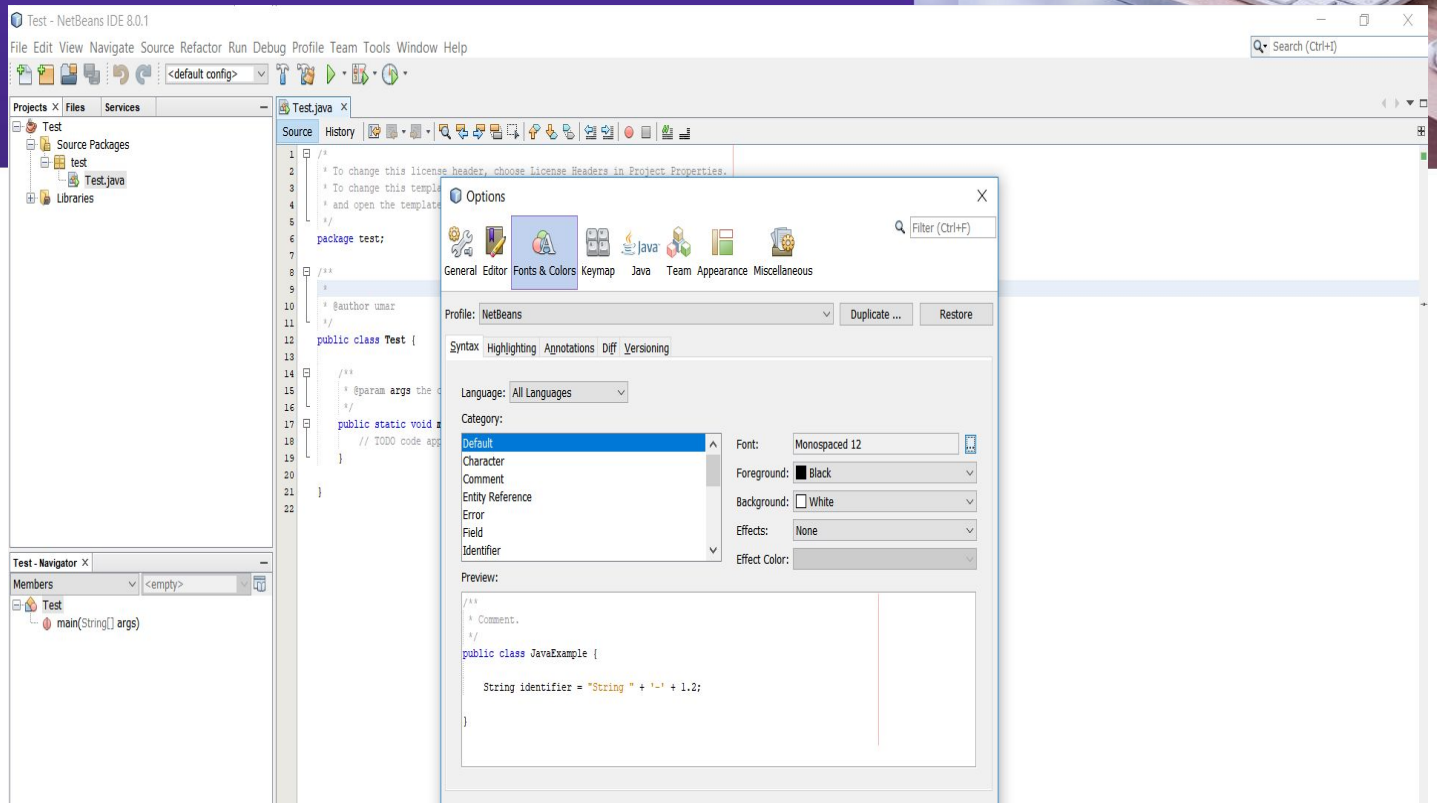

NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
New Project... Ctrl+Shift+N
New File... Ctrl+N
Open Project... Ctrl+Shift+O
Open Recent Project
Close Project
Close Other Projects
Close All Projects
Open File...
Open Recent File
Project Groups...
Project Properties
Import Project
Export Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Page Setup...
Print... Ctrl+Alt+Shift+P
Print to HTML...
Exit

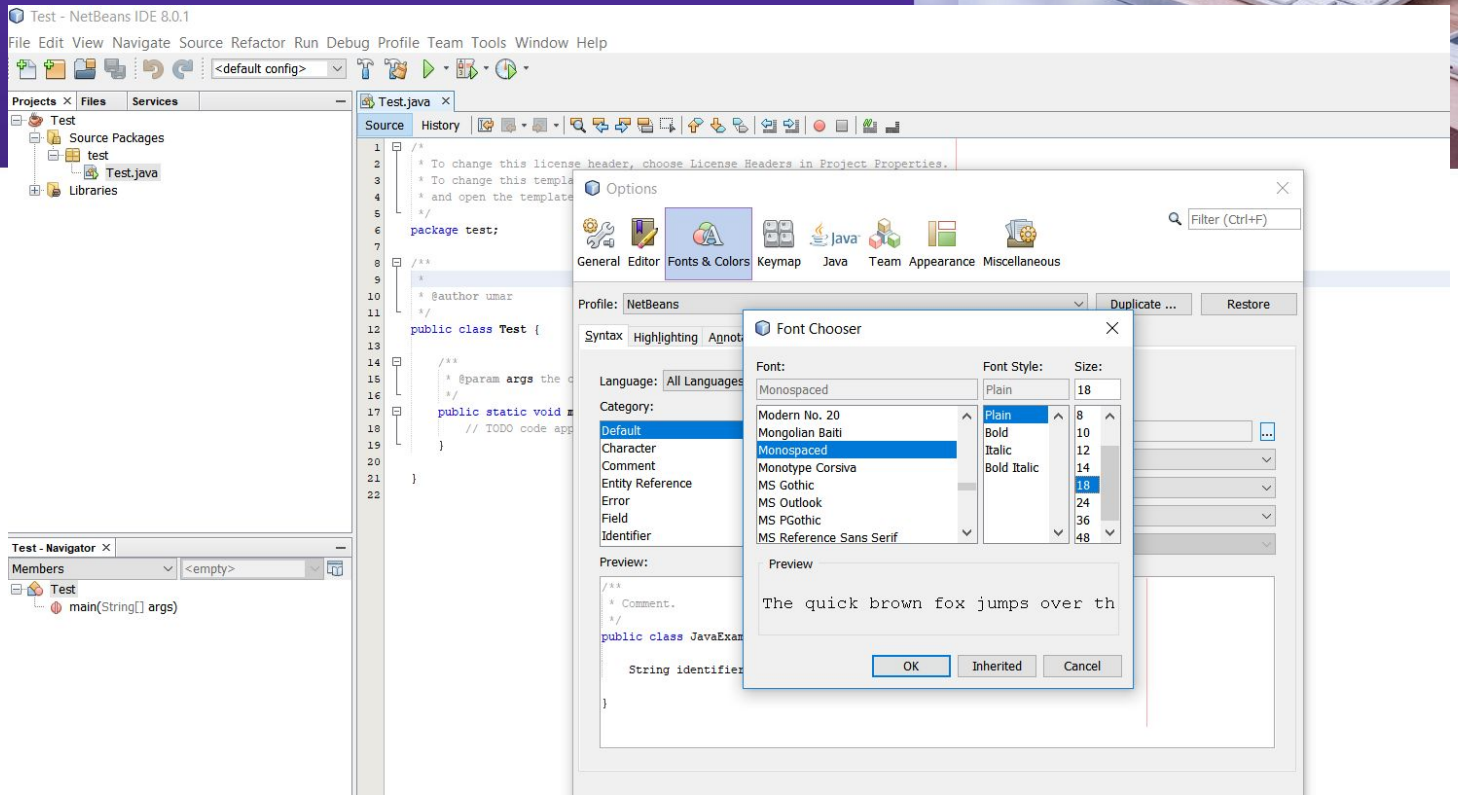

NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects
Files
Services
<No Project Open>
New Project
Steps
1. Choose Project
2. ...
Choose Project
Filter:
Categories:
Java
JavaFX
Maven
NetBeans Modules
Samples
Projects:
Java Application
Java Class Library
Java Project with Existing Sources
Java Free-Form Project
Description:
Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use an IDE-generated Ant build script to build, run, and debug your project.
< Back
Next >
Finish
Cancel
Help

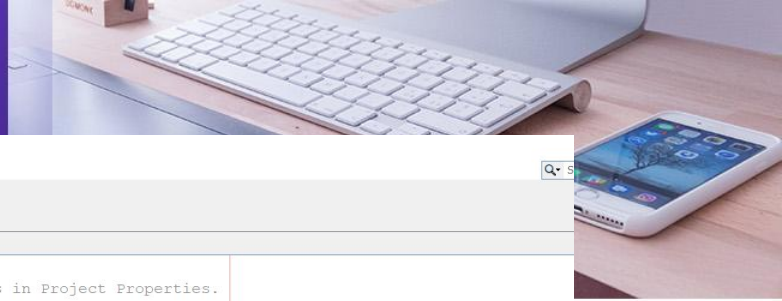

NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Files
Services
<No Project Open>
New Java Application
Steps
1. Choose Project
2. Name and Location
Name and Location
Project Name: Test
Project Location: C:\Users\umar\Documents\NetBeansProjects
Browse...
Project Folder: C:\Users\umar\Documents\NetBeansProjects\Test
Use Dedicated Folder for Storing Libraries
Libraries Folder:
Browse...
Different users and projects can share the same compilation libraries (see Help for details).
Create Main Class test.Test
< Back
Next >
Finish
Cancel
Help


Test - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects
Files
Services
Test
Source Packages
test
Test.java
Libraries
Test.java
Source
History
/*
* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/
package test;
/**
*
* @author umar
*/
public class Test {
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
// TODO code application logic here
}
}
Test - Navigator
Members
<empty>
Test
main(String[] args)


Test - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects Files Services
Test
Source Packages
test
Test.java
Libraries
Test - Navigator
Members
<empty>
Test
main(String[] args)
Test.java
Source History
Apply Diff Patch...
Diff
Add to Favorites
Create/Update Tests
Analyze Javadoc
Add to Palette...
Internationalization
Java Platforms
NetBeans Platforms
Ant Variables
Libraries
Templates
DTDs and XML Schemas
Palette
Plugins
Options


Test - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects
Files
Services
Test
Source Packages
test
Test.java
Libraries
Test.java
Source
History
package test;
public class Test {
Test - Navigator
Members
<empty>
Test
main(String[] args)
Options
Filter (Ctrl+F)
General
Editor
Fonts & Colors
Keymap
Java
Team
Appearance
Miscellaneous
Profile:
NetBeans
Duplicate ...
Restore
Syntax
Highlighting
Annotations
Diff
Versioning
Language:
All Languages
Category:
Default
Character
Comment
Entity Reference
Error
Field
Identifier
Font:
Monospaced 12
Foreground:
Black
Background:
White
Effects:
None
Effect Color:
Preview:
/**
* Comment.
*/
public class JavaExample {
String identifier = "String " + '-' + 1.2;
}


Test - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects Files Services
Test
Source Packages
test
Test.java
Libraries
Test - Navigator
Members
<empty>
Test
main(String[] args)
Test.java
Source History
/*
* To change this license header, choose License Headers in Project Properties.
* To change this templa
* and open the template
*/
package test;
/**
*
* @author umar
*/
public class Test {
/**
* @param args the c
*/
public static void m
// TODO code app
}
}
Options
General Editor Fonts & Colors Keymap Java Team Appearance Miscellaneous
Filter (Ctrl+F)
Profile: NetBeans
Duplicate ...
Restore
Syntax Highlighting Annot
Language: All Languages
Category:
Default
Character
Comment
Entity Reference
Error
Field
Identifier
Preview:
/**
* Comment.
*/
public class JavaExam
String identifier
}
Font Chooser
Font:
Monospaced
Font Style:
Plain
Size:
18
Modern No. 20
Mongolian Baiti
Monospaced
Monotype Corsiva
MS Gothic
MS Outlook
MS PGothic
MS Reference Sans Serif
Plain
Bold
Italic
Bold Italic
8
10
12
14
18
24
36
48
Preview
The quick brown fox jumps over th
OK
Inherited
Cancel



Test - NetBeans IDE 8.0.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects | Files | Services

- Test
  - Source Packages
    - test
      - Test.java
  - Libraries

Test - Navigator

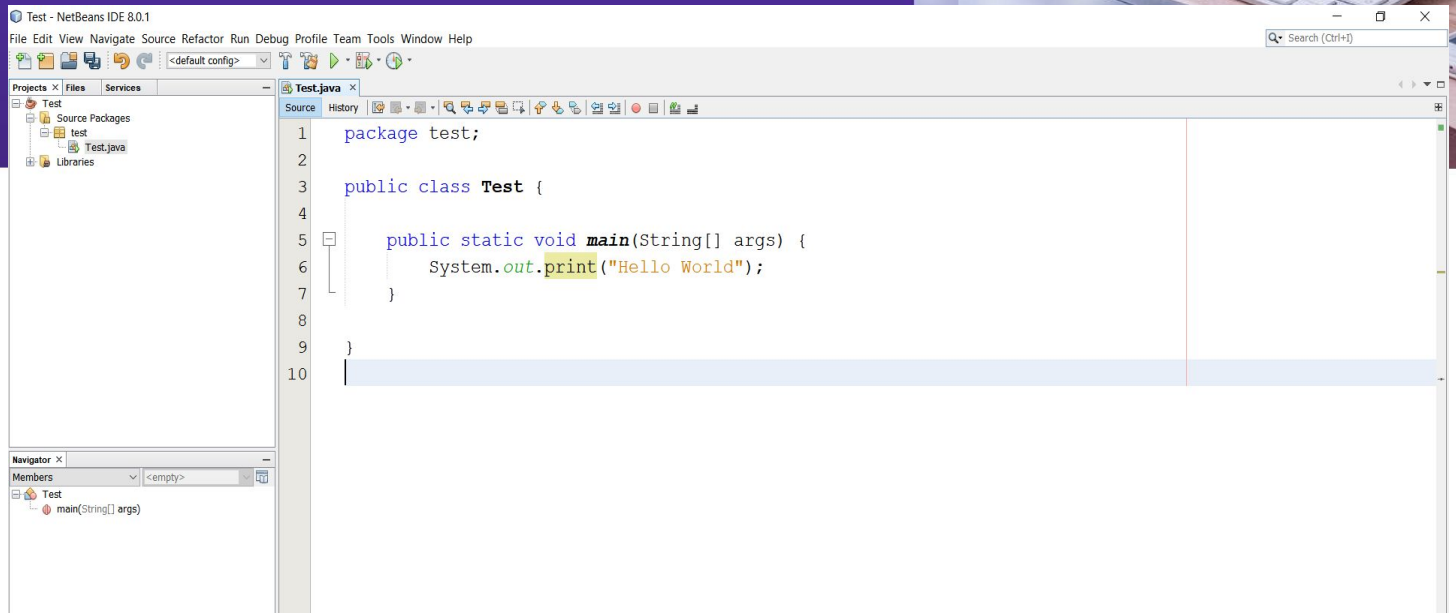
Members <empty>

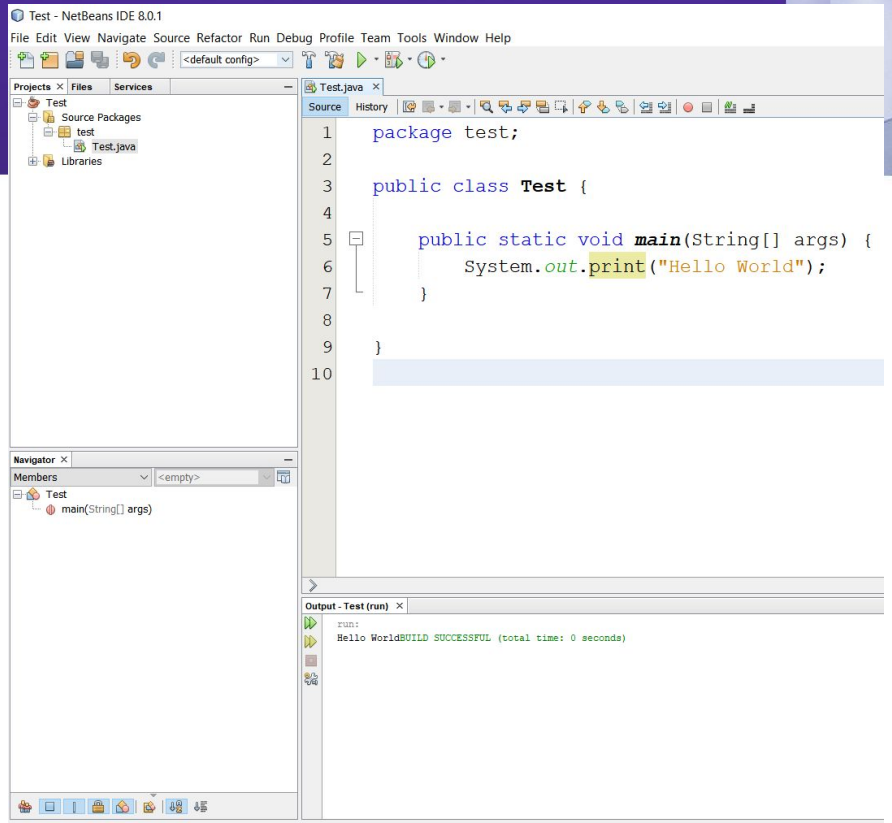
- Test
  - main(String[] args)

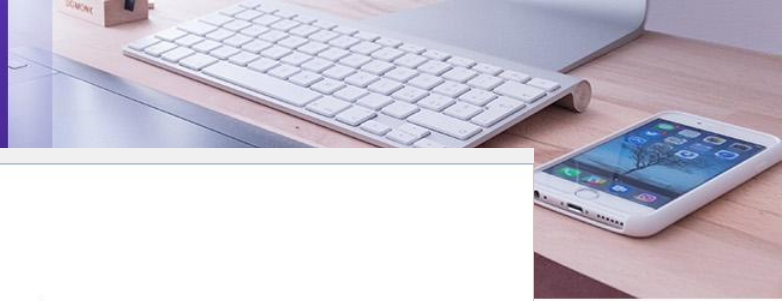
Test.java

Source | History

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package test;
7
8  /**
9   *
10  * @author umar
11  */
12 public class Test {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19     }
20
21 }
22
```


Test - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Projects
Files
Services
Test
Source Packages
test
Test.java
Libraries
Test.java
Source
History
package test;
public class Test {
public static void main(String[] args) {
System.out.print("Hello World");
}
}
Navigator
Members
<empty>
Test
main(String[] args)


Test - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects Files Services
Test
Source Packages
test
Test.java
Libraries
Navigator
Members
<empty>
Test
main(String[] args)
Test.java
Source History
package test;
public class Test {
public static void main(String[] args) {
System.out.print("Hello World");
}
}
Output - Test (run)
run:
Hello WorldBUILD SUCCESSFUL (total time: 0 seconds)



Source Packages
- test
  - Test.java
- Libraries

Source | History

```
1  package test;
2
3  public class Test {
4
5      public static void main(String[] args) {
6          System.out.print("Hello World");
7      }
8
9  }
10
```

Navigator

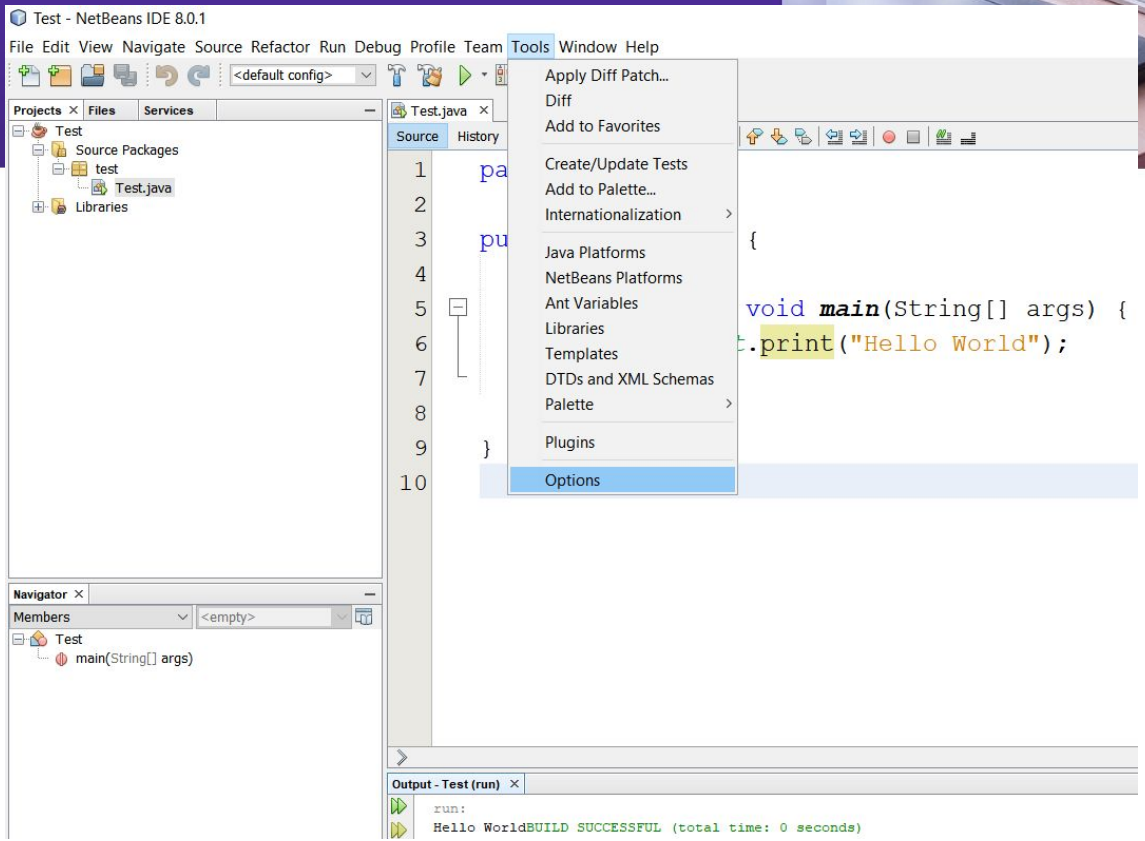
Members | <empty>

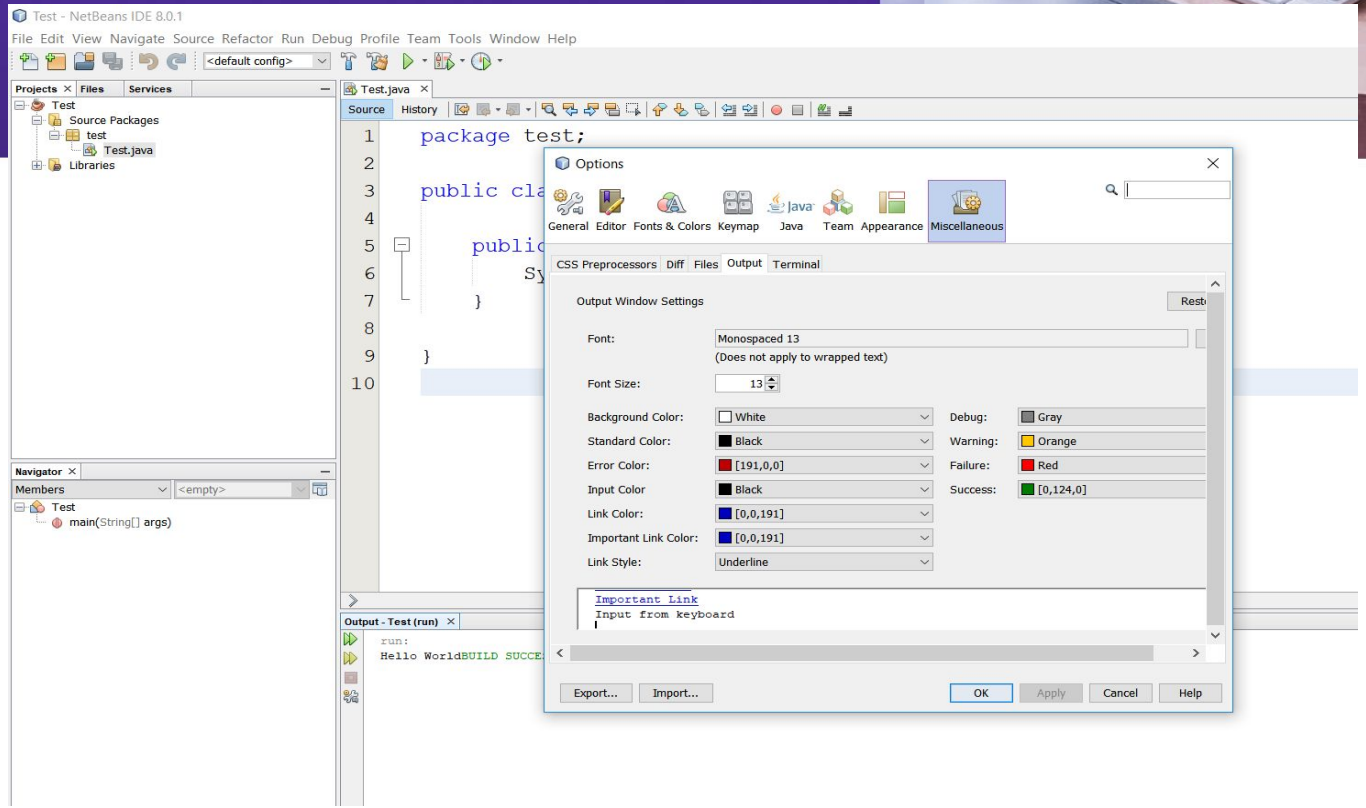
- Test
  - main(String[] args)

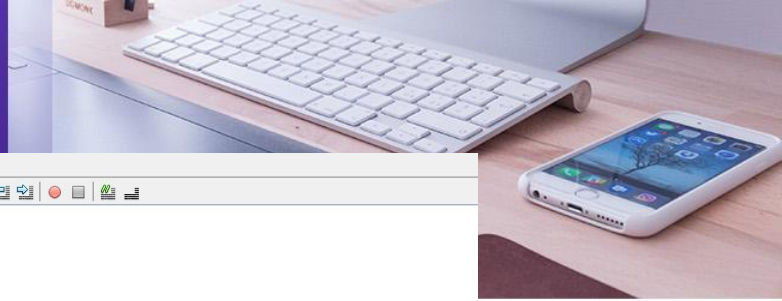
Output - Test (run)

```
run:
Hello WorldBUILD SUCCESSFUL (tota
```

Copy	Ctrl+C
Paste	Ctrl+V
Find...	Ctrl+F
Find Next	F3
Find Previous	Shift+F3
Filter...	Ctrl+G
Wrap text	Ctrl+R
Larger Font	Ctrl+Up
Smaller Font	Ctrl+Down


Test - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects Files Services
Test
Source Packages
test
Test.java
Libraries
Navigator
Members
<empty>
Test
main(String[] args)
Test.java
Source History
Apply Diff Patch...
Diff
Add to Favorites
Create/Update Tests
Add to Palette...
Internationalization
Java Platforms
NetBeans Platforms
Ant Variables
Libraries
Templates
DTDs and XML Schemas
Palette
Plugins
Options
void main(String[] args) {
.print("Hello World");
}
Output - Test (run)
run:
Hello WorldBUILD SUCCESSFUL (total time: 0 seconds)


Test - NetBeans IDE 8.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects Files Services
Test
Source Packages
test
Test.java
Libraries
Navigator
Members
<empty>
Test
main(String[] args)
Test.java
Source History
package test;
public cla
public
Sy
}
}
Output - Test (run)
run:
Hello WorldBUILD SUCCE
Options
General Editor Fonts & Colors Keymap Java Team Appearance Miscellaneous
CSS Preprocessors Diff Files Output Terminal
Output Window Settings
Rest
Font: Monospaced 13
(Does not apply to wrapped text)
Font Size: 13
Background Color: White
Standard Color: Black
Error Color: [191,0,0]
Input Color Black
Link Color: [0,0,191]
Important Link Color: [0,0,191]
Link Style: Underline
Debug: Gray
Warning: Orange
Failure: Red
Success: [0,124,0]
Important Link
Input from keyboard
Export... Import... OK Apply Cancel Help



Projects × Files Services

- Test
  - Source Packages
    - test
      - Test.java
  - Libraries

Navigator ×

Members <empty>

- Test
  - main(String[] args)

Test.java ×

Source History

```
1  package test;
2
3  public class Test {
4
5      public static void main(String[] args) {
6          System.out.print("Hello World");
7      }
8
9  }
10
```

Output - Test (run) ×

```
run:
Hello WorldBUILD SUCCESSFUL (total time: 0 seconds)
```

Programming Fundamentals
Fall 2021

Lecture 2

- Programming Basics

What is a Variable?

- **Variables** are places where information can be stored while a program is running.

- Their values can be changed at any point over the course of a program

Creating Variables

- To create a variable, **declare** its name and the type of information that it will store.
- The type is listed first, followed by the name.
- Example: a variable that stores an integer representing the highest score on an exam could be declared as follows:

```
int      highScore;
```

type name

- Now you have the variable (highScore), you will want to assign a value to it.
- Example: the highest score in the class exam is 98.

```
highScore = 98;
```

Naming Variables

- The name that you choose for a variable is called an **identifier**. In Java, an **identifier** can be of any length, but must start with:

```
a letter (a - z),
          a dollar sign ($),
          or, an underscore ( _ ).
```

- In addition, there are certain **keywords** reserved (e.g., "class") in the Java language which can *never* be used as identifiers.

Naming (Continued)

- Java is a **case-sensitive** language – the capitalization of letters in identifiers matters.
  A `rose` is not a `Rose` is not a `ROSE`

- It is good practice to select variable names that give a good indication of the sort of data they hold
  - For example, if you want to record the size of a video, `videoSize` is a good choice for a name whereas `qqq` would be a bad choice

Naming (Continued)

- When naming a variable, the following convention is commonly used:
  - The first letter of a variable name is lowercase
  - Each successive word in the variable name begins with a capital letter
  - All other letters are lowercase
- Here are some examples:

```
pageCount
loadFile
anyString
threeWordVariable
```

Statements

- A **statement** is a command that causes something to happen.
- All statements in Java are separated by semicolons ;
- Example:

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

- You have already used statements to create a variable and assign it a value.

Variables and Statements

- One way is to **declare** a variable and then **assign** a value to it with two statements:

```
int e;  // declaring a variable
e = 5;  // assigning a value to a variable
```

- Another way is to write a single **initialization** statement:

```
int e = 5;  // declaring AND assigning
```

Primitive Data Types

- There are eight built-in (primitive) data types in the Java language
  - 4 integer types (`byte`, `short`, `int`, `long`)
  - 2 floating point types (`float`, `double`)
  - Boolean (`boolean`)
  - Character (`char`)

Integer Data Types

Data Type	Value Range
`byte`	**-128 to +127**
`short`	**-32768 to +32767**
`int`	**-2147483648 to +2147483647**
`long`	**-9223372036854775808 to +9223372036854775807**

- There are four data types that can be used to store integers.
- The one you choose to use depends on the size of the number that we want to store.
- In this course, we will always use `int` when dealing with integers.

Example

Helloworld.java

```
1
2  public class Helloworld {
3
4      public static void main(String[] args) {
5          byte a;
6          a=127;
7
8
9      }
10
11 }
12
13
```

Problems | @ Javadoc | Declaration

0 errors, 1 warning, 0 others

Description	Resource	Path
Warnings (1 item)		
The value of the local variable a is not used	Helloworld.java	/Test/src

Example

Helloworld.java

```
1
2  public class Helloworld {
3
4      public static void main(String[] args) {
5          byte a;
6          a=128;
7
8
9      }
10
11 }
12
13
```

Problems | @ Javadoc | Declaration

1 error, 0 warnings, 0 others

Description	Resource
Errors (1 item)	

Examples


```
public class Helloworld {

public static void main(String[] args) {

          int a=5;
          System.out.println("The value of a is"+a);

                                    }
}
```

Examples

```
public class Helloworld {

public static void main(String[] args) {
                                int a=5;
                                int b= 10;

              System.out.println("The value of a is"+a+b);

                              }

}
```

Ans : The value of a is 510

Examples

```
public class Helloworld {

public static void main(String[] args) {
                                int a=5;
                                int b= 10;

                            System.out.println(a+b);

                            }

}
```

Ans :15

Examples

```
public class Helloworld {

public static void main(String[] args) {
                                int a=5;
                                int b= 10;
                                int sum = a+b;
              System.out.println("The sum of a and b "+sum);

                            }

}
```

The sum of a and b 15

Floating Point Data Types

- There are two data types that can be used to store decimal values (real numbers).
- The one you choose to use depends on the size of the number that we want to store.
- *float* a= 2.3f;
- *Double*

Integer Division Example

- ```
  int i = 63;
  int j = 35;
  System.out.println(i / j);
  Output: 1
  ```

- ```
  double x = 63;
  double y = 35;
  System.out.println(x / y);
  Ouput: 1.8
  ```

- The result of integer division is just the integer part of the quotient!

Boolean Data Type

- Boolean is a data type that can be used in situations where there are two options, either **`true`** or **`false`**.

- Example:

```
boolean Hungry = true;
boolean fileOpen = false;
```

Character Data Types

- Character is a data type that can be used to store a single characters such as a letter, number, punctuation mark, or other symbol.

- Example:
  - `char firstLetterOfName = 'e' ;`
  - `char myQuestion = '?' ;`

- Note that you need to use singular quotation marks when assigning `char` data types.

Introduction to Strings

- Strings consist of a series of characters inside double quotation marks.

- Examples statements assign `String` variables:

```
String Author = “abc";
String password = “123";
```

Reserved Words

abstract	assert	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	violate	while	

Important links

- My email: farhan.aadil@ciit-attock.edu.pk
- My web page: DrAadil.com

Thanks for listening!
Ready for Questions & Answers
☺️