**Practical -1**

**Aim: Installation of NetBeans on Windows.**

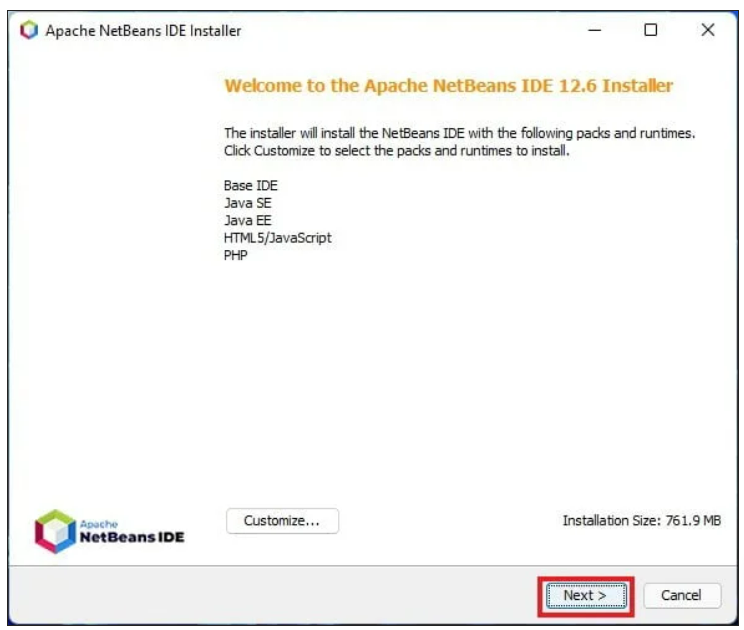
you have installed Java JDK on your system, you can follow these steps to install and set up NetBeans IDE for Java Programming. You can also check our list of the [6 best IDE for Java Programming Language](https://www.thecoderworld.com/best-ide-for-java-programming-language/). Let’s check the steps to install NetBeans:

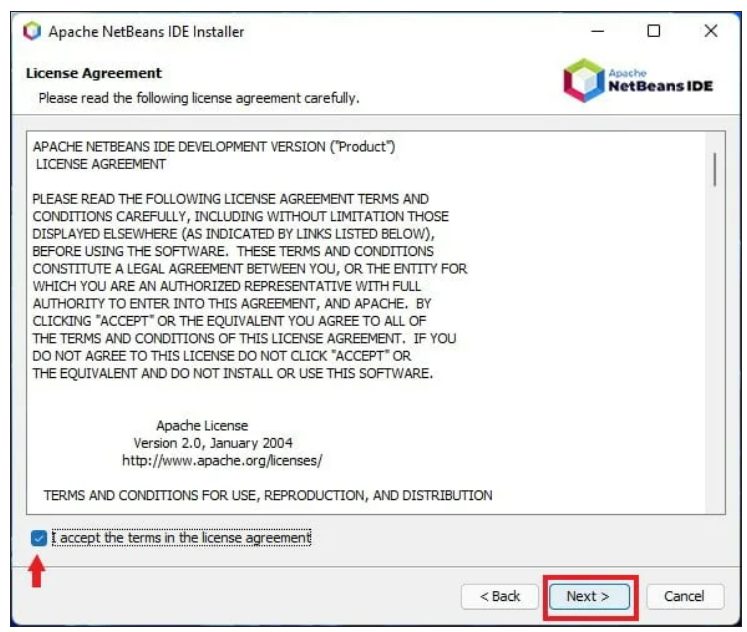
**Step 1:** Open the browser and search for NetBeans and open the link from [netbeans.apache.org](https://netbeans.apache.org/)

**Step 2:** It will redirect you to the official web page. Click the “**Download**” button and select the [windows installer](https://netbeans.apache.org/download/nb16/) file to download Apache NetBeans IDE.

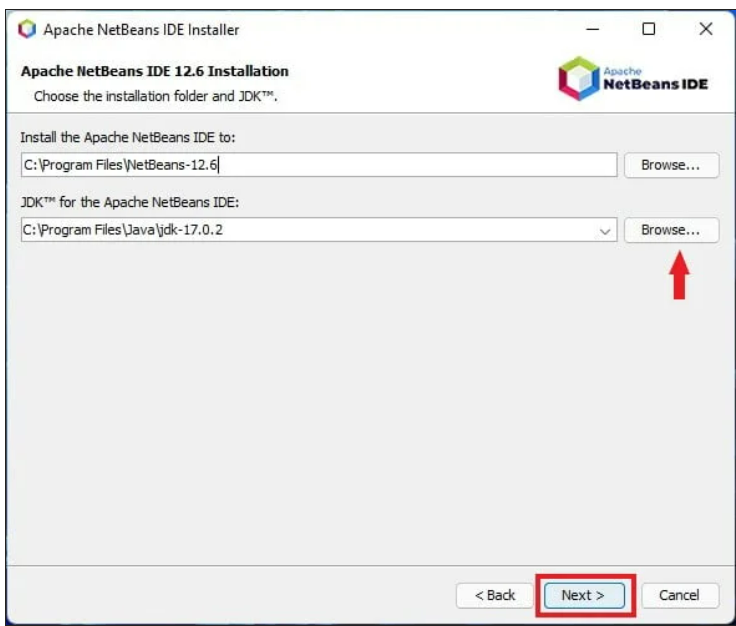


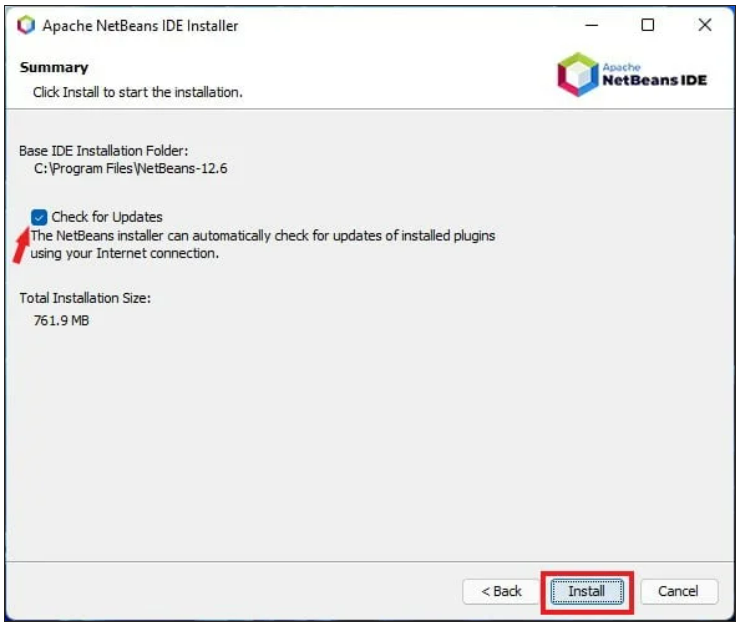
**Step 3**: After the download is complete, open the downloaded file. Now, it will open Apache NetBeans IDE Installer window, here click on “**Next**” to continue.



**Step 4:** In the next window, you will find the license agreement. Read the agreement and agree to the terms of the license agreement. After that, click on the “Next”button.

**Step 5**: Now, you will get an option to change the installation windows. You can click on the “**Browse**” option if you want to change the installation location. Or else, you can continue with the default location and click on “**Next**” to continue. Here you also need to add the Java JDK installation location. After that, click on “**Next**” to continue.



**Step 6**: On the next window, select check for updates and click the “**Install**” button. The installation will take some time, so wait until the installation is complete. After that, you are ready to use NetBeans for Java Development

That’s it; this is how you can install Apache NetBeans IDE on Windows 11. Now let’s discuss how to set up NetBeans for Java development.

**Practical :- 02**

**Aim :-** WAP to demonstrate life cycle of Applet.

**Life Cycle Of Applet :-**

The life cycle of applet mainly consists of five different methods that are :-

* **Initialization:**
  + The init() method is called once when the applet is first loaded.
  + Used for initialization tasks such as setting up resources.
* **Starting:**
  + The start() method is called each time the applet becomes active.
  + Handles tasks like starting animations or threads.
* **Stopping:**
  + The stop() method is invoked when the applet is no longer active.
  + Used to pause animations or stop threads.
* **Destruction:**
  + The destroy() method is called when the applet is about to be unloaded.
  + Allows for cleanup activities like releasing resources.
* **Painting:**
  + The paint(Graphics g) method is called whenever the applet needs to redraw its content.
  + Automatically invoked after init() and start(), and whenever the applet is resized or needs to be repainted.
  + Used to perform custom drawing and rendering, such as drawing shapes, text, or images on the applet's window.
  + Can be explicitly called by repaint() method to refresh the applet's display.

Top of Form

Bottom of Form

# Practical :-

import java.applet.Applet;

import java.awt.Graphics;

import java.awt.Color;

/\*\*

\* Practical 2 : WAP to demonstrate life cycle of Applet.

\* @author Rudra Mistry

\* Division : Q-C

\* Batch : Q-C1

\*/

public class Practical2 extends Applet {

public void init() {

super.init();

System.out.println("hello.init()");

}

public void start() {

super.start();

System.out.println("hello.start()");

}

public void paint(Graphics g){

g.setColor(Color.red);

g.drawString("Hello World", 100, 100);

}

public void stop() {

super.stop();

System.out.println("hello.stop()");

}

public void destroy() {

super.stop();

System.out.println("hello.desttroy()");

}

#OUTPUT

