Week-1

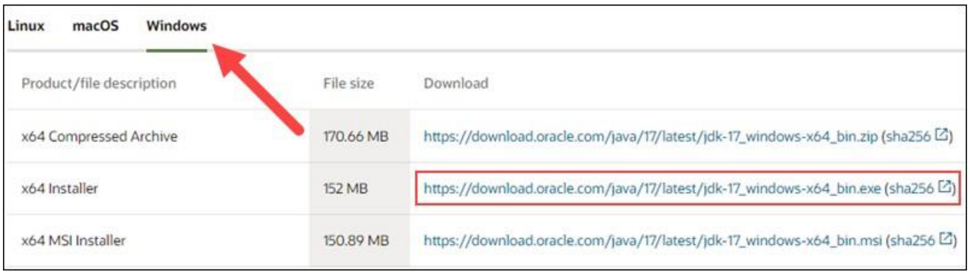
Aim: Install and setup java environment, configure workspace to execute java program.

Experiment-1.1: Install and Setup Java Environment

Download Java for Windows10

1) Download the latest Java Development Kit installation file for Windows 10 to have the latest features and bug fixes.

2) On the Downloads page, click the x64Installer download link under the Windows category. At the time of writing this article, Java version17 is the latest long- term support Java version. Wait for the download to complete.



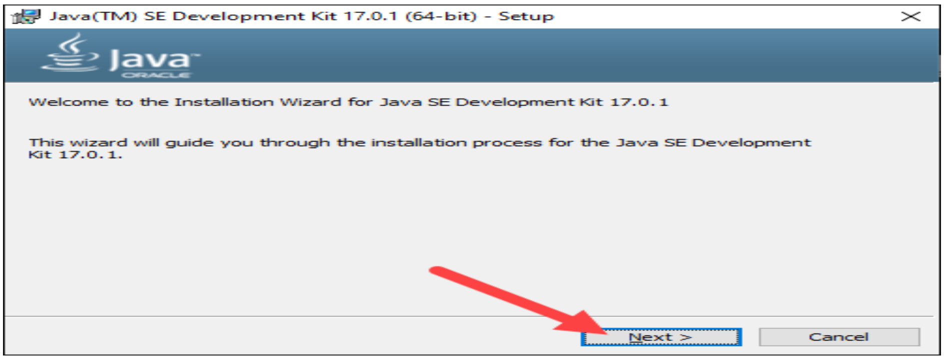
Product/file description	File size	Download
x64 Compressed Archive	170.66 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip (sha256)
x64 Installer	152 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe (sha256)
x64 MSI Installer	150.89 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi (sha256)

- InstallJavaonWindows10

1) After downloading the installation file, proceed with installing Java on your Windows system.

Step1:RuntheDownloadedFile.Double-clickthedownloadedfiletostarttheinstallation.

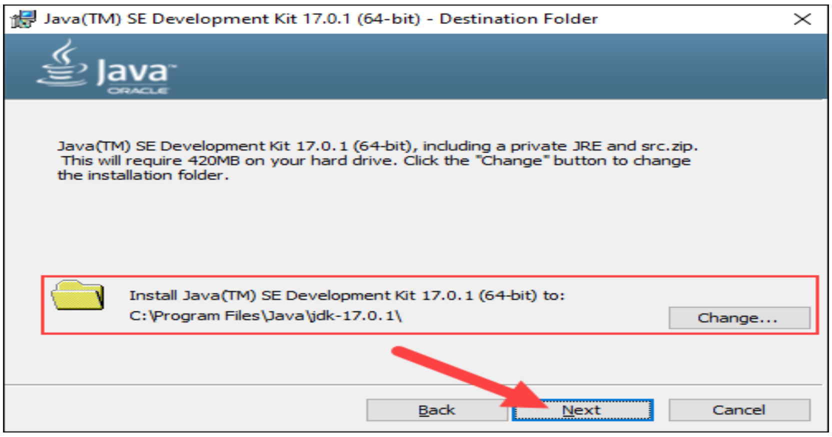
Step2: Configure the Installation Wizard. After running the installation file, the installation wizard welcome screen appears. Click Next to proceed to the next step.


Java(TM) SE Development Kit 17.0.1 (64-bit) - Setup

Welcome to the Installation Wizard for Java SE Development Kit 17.0.1

This wizard will guide you through the installation process for the Java SE Development Kit 17.0.1.

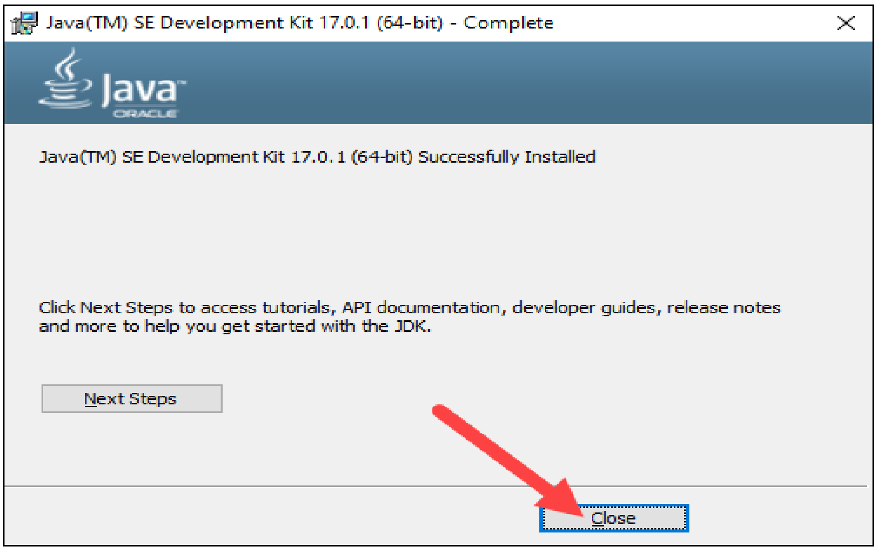
Choose the destination folder for the Java installation files or stick to the default path. Click Next to proceed.


Java(TM) SE Development Kit 17.0.1 (64-bit) - Destination Folder

Java(TM) SE Development Kit 17.0.1 (64-bit), including a private JRE and src.zip. This will require 420MB on your hard drive. Click the "Change" button to change the installation folder.

Install Java(TM) SE Development Kit 17.0.1 (64-bit) to:
C:\Program Files\Java\jdk-17.0.1\

Wait for the wizard to finish the installation process until the Successfully Installed message appears. Click Close to exit the wizard.


Java(TM) SE Development Kit 17.0.1 (64-bit) - Complete

Java(TM) SE Development Kit 17.0.1 (64-bit) Successfully Installed

Click Next Steps to access tutorials, API documentation, developer guides, release notes and more to help you get started with the JDK.

Experiment-1.2: Install java editor (Eclipse for Enterprise Java) and configure workspace

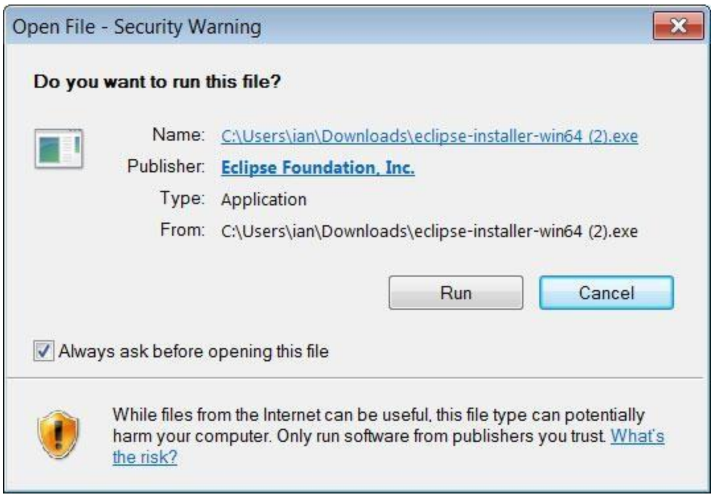
Steps:

1) Download the Eclipse Installer

DownloadEclipseInstallerfromhttp://www.eclipse.org/downloads

2) Start the Eclipse Installer executable

For Windows users, after the Eclipse Installer executable has finished downloading it should be available in your download directory. Start the Eclipse Installer executable. You may get a security warning to run this file. If the Eclipse Foundation is the Publisher, you are good to select Run.



Open File - Security Warning

Do you want to run this file?

Name: C:\Users\ian\Downloads\eclipse-installer-win64 (2).exe

Publisher: Eclipse Foundation, Inc.

Type: Application

From: C:\Users\ian\Downloads\eclipse-installer-win64 (2).exe

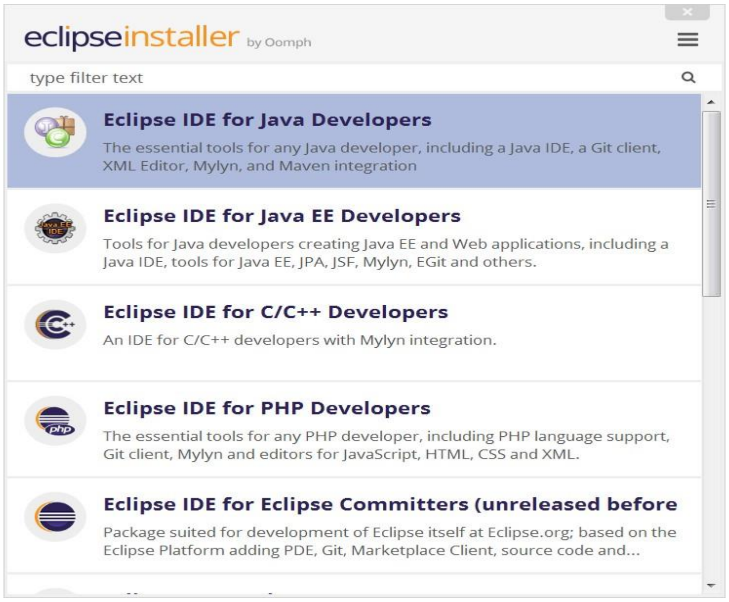
☑ Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

3. Select the package to install

The new Eclipse Installer shows the packages available to Eclipse users. You can search for the package you want to install or scroll through the list

Select and click on the package you want to install.



Eclipse IDE for Java Developers
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, and Maven integration

Eclipse IDE for Java EE Developers
Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn, EGit and others.

Eclipse IDE for C/C++ Developers
An IDE for C/C++ developers with Mylyn integration.

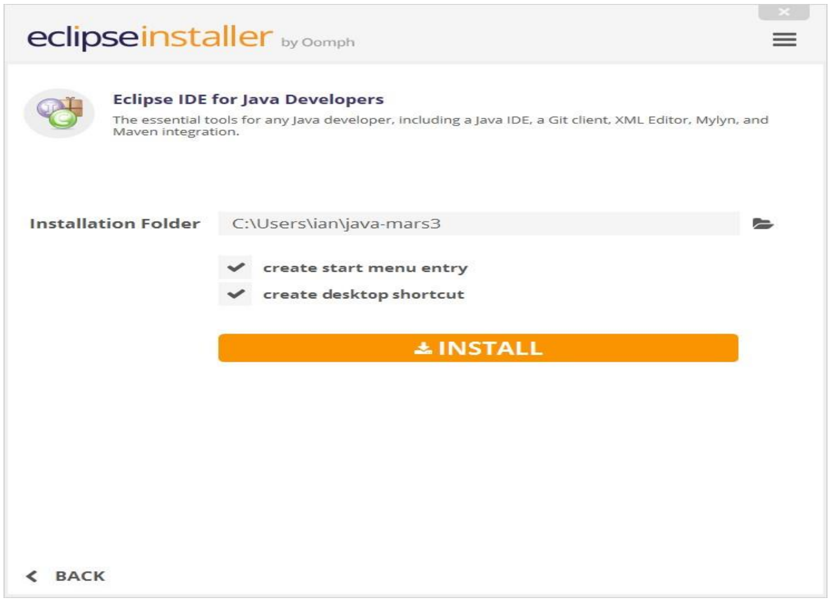
Eclipse IDE for PHP Developers
The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.

Eclipse IDE for Eclipse Committers (unreleased before
Package suited for development of Eclipse itself at Eclipse.org; based on the Eclipse Platform adding PDE, Git, Marketplace Client, source code and...

4. Select your installation folder

Specify the folder where you want Eclipse to be installed. The default folder will be in your User directory.

Select the 'Install' button to begin the installation.



Eclipse IDE for Java Developers
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, and Maven integration.

Installation Folder: C:\Users\ian\java-mars3

✓ create start menu entry
✓ create desktop shortcut

5. Launch Eclipse

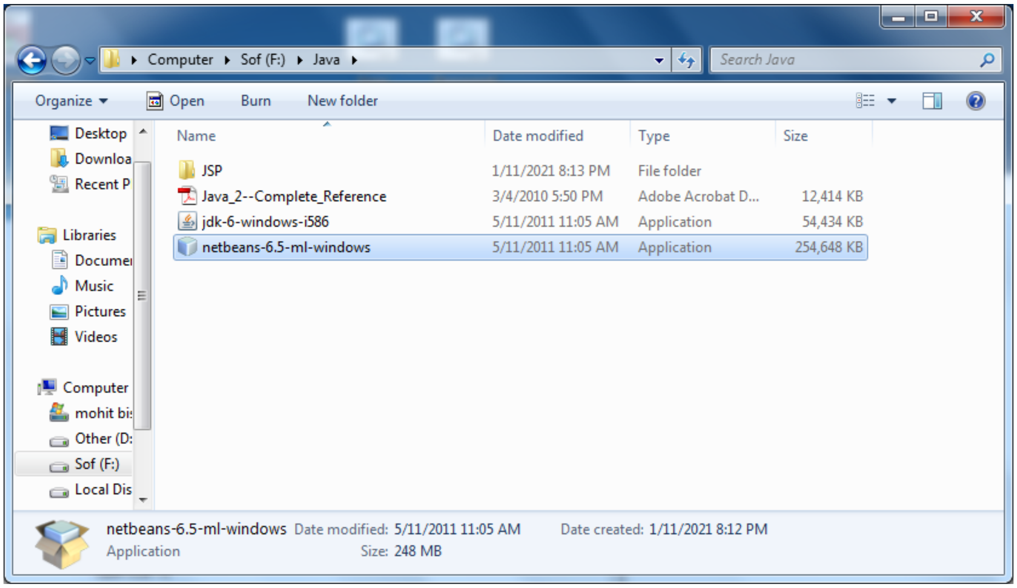
Once the installation is complete you can now launch Eclipse.

Experiment-1.2: Install java editor (NetBeans for Enterprise Java) and configure workspace

NetBeans: NetBeans IDE is a Free open-Source, Cross-plate form Integrated Development Environment (IDE) with built-in support for the JAVA Programming Language. It can run any machine which consists of the Java Virtual Machine (JVM).

Steps:

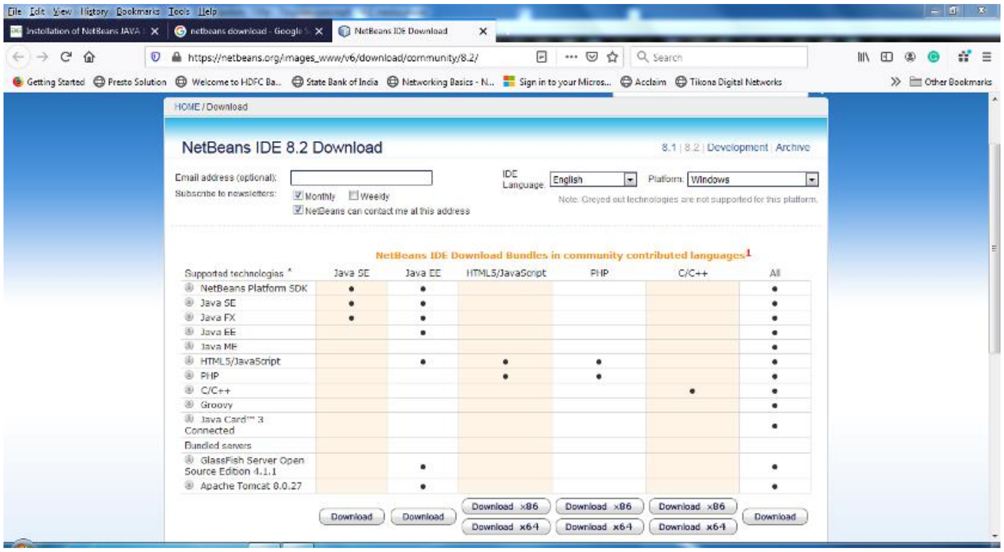
1. You need to have a setup file of the NetBeans JAVA into your setup.



Name	Date modified	Type	Size
JSP	1/11/2021 8:13 PM	File folder	
Java_2--Complete_Reference	3/4/2010 5:50 PM	Adobe Acrobat D...	12,414 KB
jdk-6-windows-i586	5/11/2011 11:05 AM	Application	54,434 KB
netbeans-6.5-ml-windows	5/11/2011 11:05 AM	Application	254,648 KB

netbeans-6.5-ml-windows Date modified: 5/11/2011 11:05 AM Date created: 1/11/2021 8:12 PM
Application Size: 248 MB

2. If you didn't have the setup you can download from the following link: https://netbeans.org/images_www/v6/download/community/8.2



File Edit View History Bookmarks Tools Help
Installation of NetBeans JAVA
netbeans download - Google S
NetBeans IDE Download
https://netbeans.org/images_www/v6/download/community/8.2/
Search
Getting Started
Presto Solution
Welcome to HDFC Ba...
State Bank of India
Networking Basics - N...
Sign in to your Micros...
Acclaim
Tikona Digital Networks
Other Bookmarks
HOME / Download
8.1 | 8.2 | Development | Archive

NetBeans IDE 8.2 Download

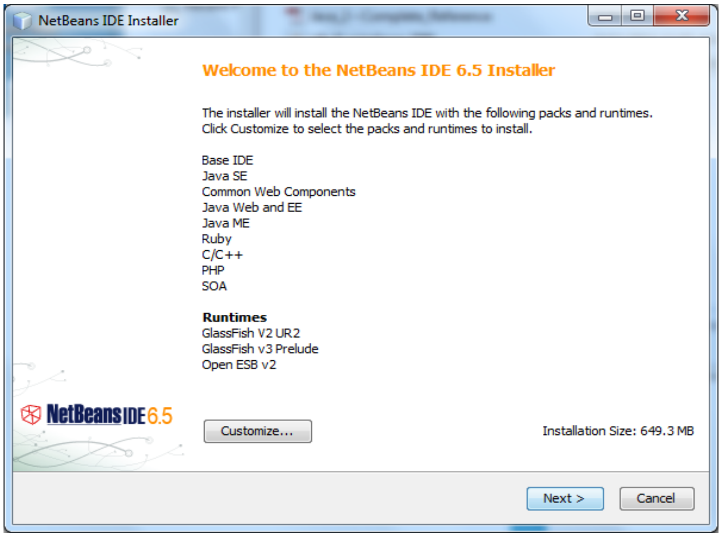
Email address (optional):
Subscribe to newsletters: ☑ Monthly ☐ Weekly
☑ NetBeans can contact me at this address

IDE Language: English
Platform: Windows
Note: Greyed out technologies are not supported for this platform.

NetBeans IDE Download Bundles in community contributed languages[1]

Supported technologies *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	All
NetBeans Platform SDK	●	●				●
Java SE	●	●				●
Java FX	●	●				●
Java EE		●				●
Java ME						●
HTML5/JavaScript		●	●	●		●
PHP			●	●		●
C/C++					●	●
Groovy						●
Java Card™ 3 Connected						●
Bundled servers						
GlassFish Server Open Source Edition 4.1.1		●				●
Apache Tomcat 8.0.27		●				●
	Download	Download	Download x86 Download x64	Download x86 Download x64	Download x86 Download x64	Download

3. You can download any type of setup as per your requirements from the above mention web page.
4. Right-click on the setup or you can Double-Click on the setup by using the mouse.
5. Click on the next option



Welcome to the NetBeans IDE 6.5 Installer

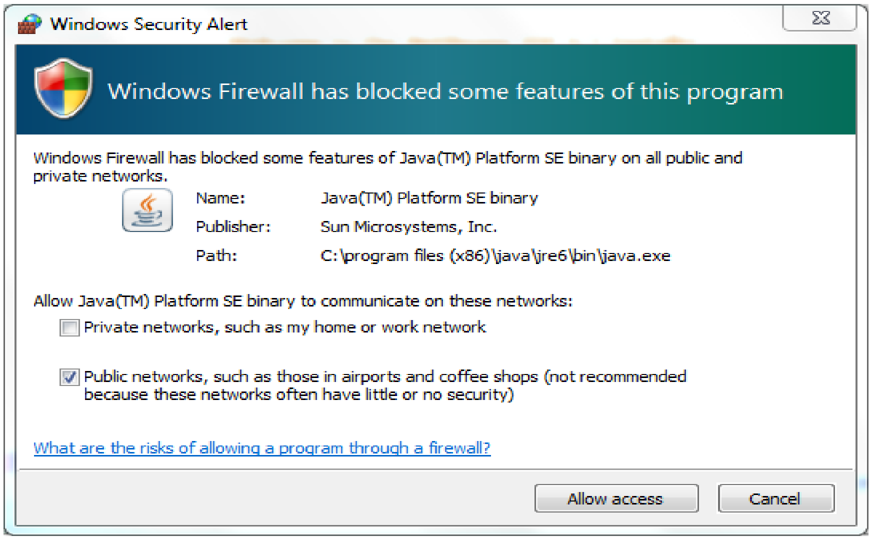
The installer will install the NetBeans IDE with the following packs and runtimes.
Click Customize to select the packs and runtimes to install.

Base IDE
Java SE
Common Web Components
Java Web and EE
Java ME
Ruby
C/C++
PHP
SOA

Runtimes
GlassFish V2 UR2
GlassFish v3 Prelude
Open ESB v2

Installation Size: 649.3 MB

6. Check on the "**Private networks, such as my home and work network".**



Windows Firewall has blocked some features of this program

Windows Firewall has blocked some features of Java(TM) Platform SE binary on all public and private networks.

Name:	Java(TM) Platform SE binary
Publisher:	Sun Microsystems, Inc.
Path:	C:\program files (x86)\java\jre6\bin\java.exe

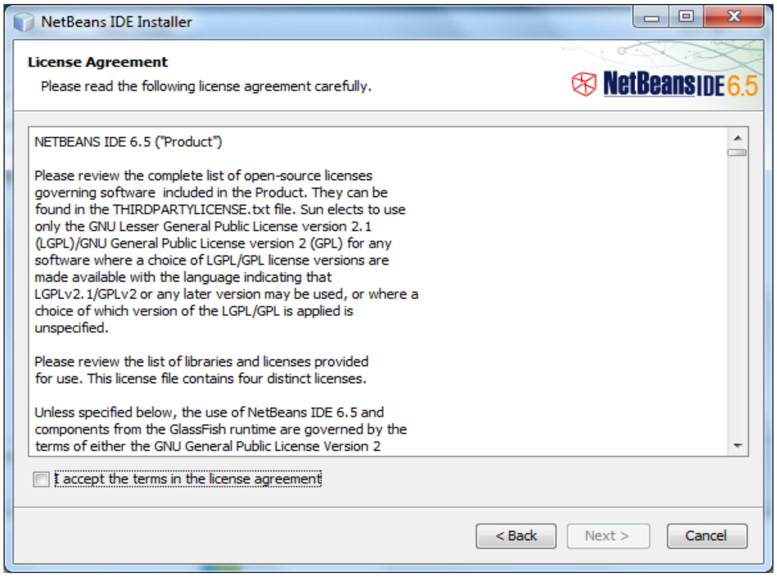
Allow Java(TM) Platform SE binary to communicate on these networks:

☐ Private networks, such as my home or work network

☑ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

What are the risks of allowing a program through a firewall?

7. Click on the **Allow access** button.
8. Check on the **"I accept"** option and click on the "**Next"** button.



License Agreement

Please read the following license agreement carefully.

NETBEANS IDE 6.5 ("Product")

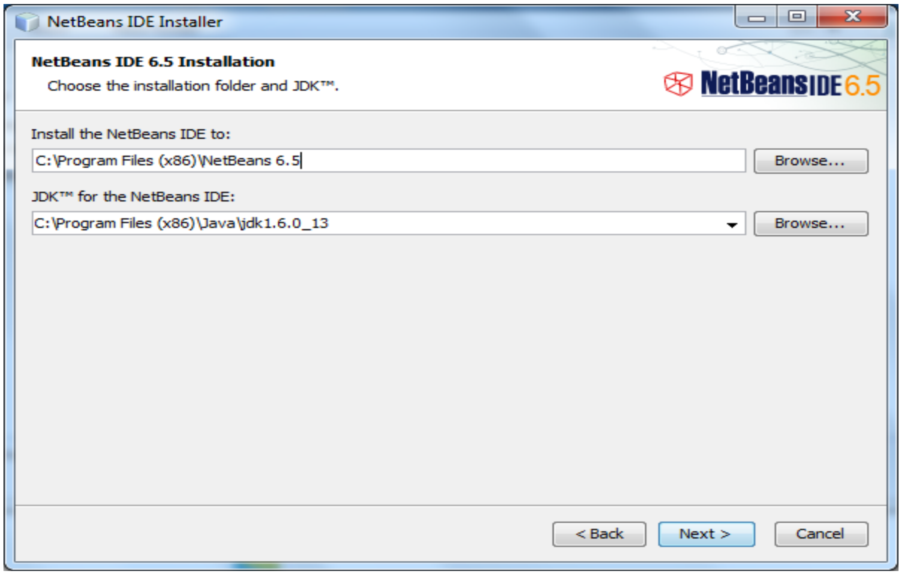
Please review the complete list of open-source licenses governing software included in the Product. They can be found in the THIRDPARTYLICENSE.txt file. Sun elects to use only the GNU Lesser General Public License version 2.1 (LGPL)/GNU General Public License version 2 (GPL) for any software where a choice of LGPL/GPL license versions are made available with the language indicating that LGPLv2.1/GPLv2 or any later version may be used, or where a choice of which version of the LGPL/GPL is applied is unspecified.

Please review the list of libraries and licenses provided for use. This license file contains four distinct licenses.

Unless specified below, the use of NetBeans IDE 6.5 and components from the GlassFish runtime are governed by the terms of either the GNU General Public License Version 2

□ I accept the terms in the license agreement

9. Select the path where you want to install the software and press the "**Next**" button.



NetBeans IDE 6.5 Installation

Choose the installation folder and JDK™.

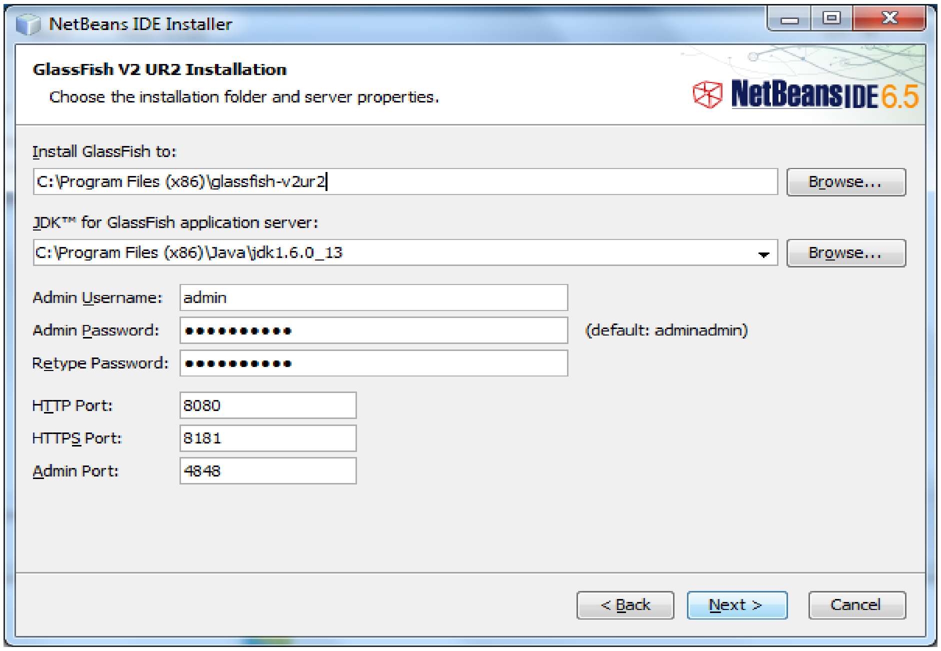
Install the NetBeans IDE to:

C:\Program Files (x86)\NetBeans 6.5

JDK™ for the NetBeans IDE:

C:\Program Files (x86)\Java\jdk1.6.0_13

10. Set the **Password, User Name & Ports** for the Network Connectivity, or we can use this **UserName and the Password** for the Connecting the Front-End to the Back-End.


NetBeans IDE Installer

GlassFish V2 UR2 Installation

Choose the installation folder and server properties.

Install GlassFish to: C:\Program Files (x86)\glassfish-v2ur2

JDK™ for GlassFish application server: C:\Program Files (x86)\Java\jdk1.6.0_13

Admin Username: admin

Admin Password: ●●●●●●●●●● (default: adminadmin)

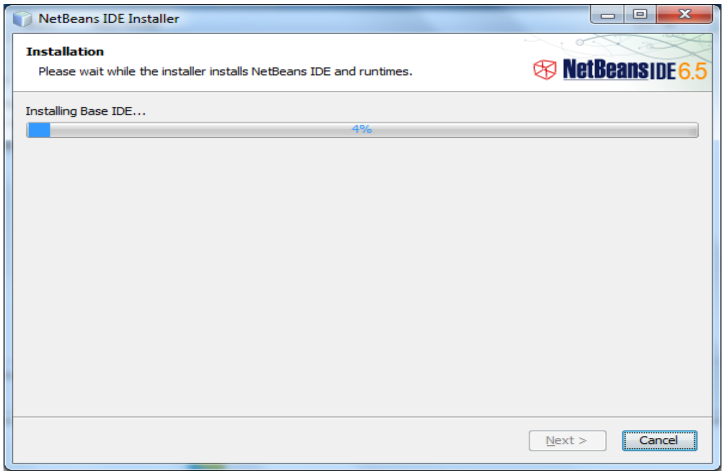
Retype Password: ●●●●●●●●●●

HTTP Port: 8080

HTTPS Port: 8181

Admin Port: 4848

11. Click on the "**Next**" button.


NetBeans IDE Installer

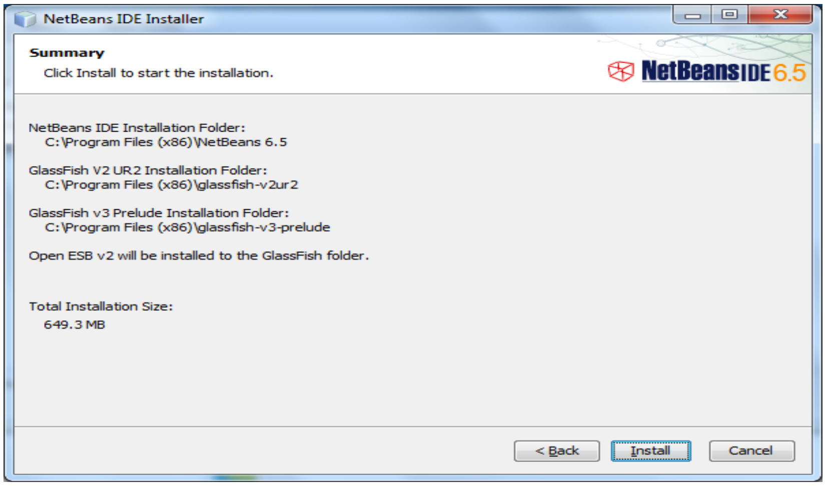
Installation

Please wait while the installer installs NetBeans IDE and runtimes.

Installing Base IDE...

4%

12. Click on the "**Install**" button.



Summary

Click Install to start the installation.

NetBeans IDE Installation Folder:
C:\Program Files (x86)\NetBeans 6.5

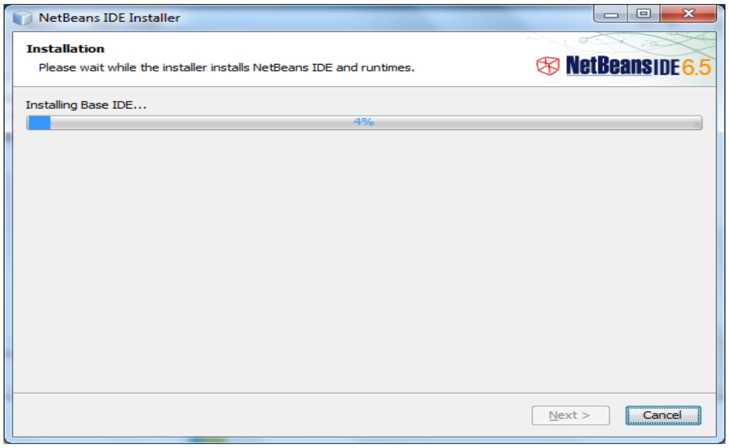
GlassFish V2 UR2 Installation Folder:
C:\Program Files (x86)\glassfish-v2ur2

GlassFish v3 Prelude Installation Folder:
C:\Program Files (x86)\glassfish-v3-prelude

Open ESB v2 will be installed to the GlassFish folder.

Total Installation Size:
649.3 MB

13. Wait for the while till the time the setup is properly installed into the Computer

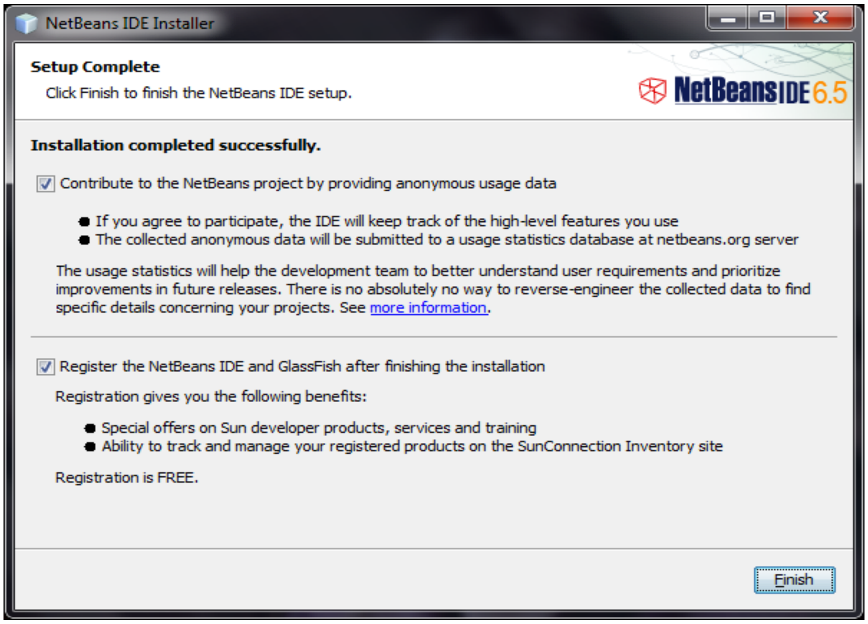


Installation

Please wait while the installer installs NetBeans IDE and runtimes.

Installing Base IDE...

14. After complication of the setup you can click on the "**Finish**" button or you can also register the Software, for Further Assistance because it is a Free Software.



Setup Complete

Click Finish to finish the NetBeans IDE setup.

Installation completed successfully.

☑ Contribute to the NetBeans project by providing anonymous usage data

- If you agree to participate, the IDE will keep track of the high-level features you use
- The collected anonymous data will be submitted to a usage statistics database at netbeans.org server

The usage statistics will help the development team to better understand user requirements and prioritize improvements in future releases. There is no absolutely no way to reverse-engineer the collected data to find specific details concerning your projects. See more information.

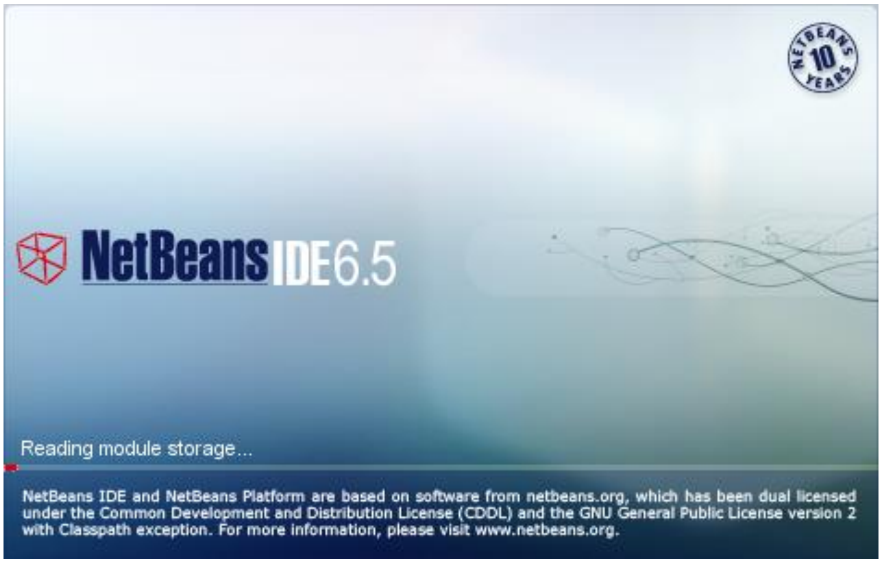
☑ Register the NetBeans IDE and GlassFish after finishing the installation

Registration gives you the following benefits:

- Special offers on Sun developer products, services and training
- Ability to track and manage your registered products on the SunConnection Inventory site

Registration is FREE.

15. Now you can start the NetBeans for further use



Reading module storage...

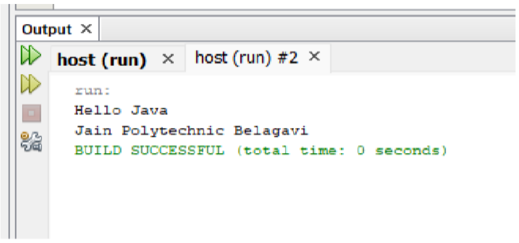
NetBeans IDE and NetBeans Platform are based on software from netbeans.org, which has been dual licensed under the Common Development and Distribution License (CDDL) and the GNU General Public License version 2 with Classpath exception. For more information, please visit www.netbeans.org.

Experiment -1.3: Write a java program to print “Hello Java” and “Jain Polytechnic Belagavi”.

Code:-

```
package javaprogram;
public class java
{
   public static void main(String[] args)
   {
          System.out.println("Hello Java");
          System.out.println("Jain Polytechnic Belagavi");
   }
}
```

Output:



```
run:
Hello Java
Jain Polytechnic Belagavi
BUILD SUCCESSFUL (total time: 0 seconds)
```

Week-2

Aim – Code, execute and debug programs that uses different types of variables and datatypes.

Experiment – 2.1: Write a java program to illustrate java variables example: (Add two numbers)

Code:-

```
package javaprogram;
public class Addition
 {
        public static void main(String[] args)
         {
                int a=10;
                int b=20;
                int c=a+b;
                System.out.println(“ Addition of “ +a+ ”And“ +b+ ” =” +c);
        }
}
```

Output:

```
Addition of 10 and 20 = 30
```

Experiment – 2.2: Write a java program to display the static and non-static properties of student class.

Code:-

```
package javaprogram;
public class Student
{
      static String name="ABC";
      static int Id=12;
      String location="Belgaum";
      char grade='A';
      public static void main(String[] args)
      {
         Student s=new Student();
          System.out.println("Name : "+Student.name+"\n"+"ID : "+Student.Id+"\n"+
                                    "Location:"+s.location+"\n"+"Grade : "+s.grade);
       }
 }
```

Output:

```
Name: ABC

ID: 12

Location: Belgaum

Grade: A
```

Experiment – 2.3: Program to illustrate creating variables of Different types

Code:

```
package mypack;
class Employee
{
	int id;
	String name;
	float Salary;

	void insert(int i, String n, float s)
	{
		id=i;
		name=n;
		Salary= s;
	}
	void display( )
	{
		System.out.println(id +" " +name+ " " +Salary);
	}
}
public class TestEmployee {

	public static void main(String[] args) {
		Employee e1 = new Employee();
		Employee e2 = new Employee();
		Employee e3 = new Employee();
		e1.insert(101, "Sudha", 45000);
		e2.insert(102, "Madhu", 55000);
		e3.insert(103, "Hari", 65000);
		e1.display();
		e2.display();
		e3.display();
	}
}
```

Output:

```
101 Sudha 45000.0
102 Madhu 55000.0
103 Hari 65000.0
```

Experiment – 2.4: Program to illustrate creating multiple objects by one type only

Code:

```
package mypack;
class Rectangle
{
        int length;
        int width;
        void insert(int l, int w)
        {
                length=l;
                width=w;
        }
        void calculateArea()
        {
                System.out.println("Area of Rectangle="+length*width);
        }
}
public class TestRectangle
 {
        public static void main(String[] args)
        {
                Rectangle r1= new Rectangle();
                Rectangle r2= new Rectangle();
                r1.insert(11, 2);
                r2.insert(3, 15);
                r1.calculateArea();
                r2.calculateArea();
        }

}
```

Output:

```
        Area of Rectangle=22
        Area of Rectangle=45
```

Experiment – 2.5 : Write a java program to achieve Method Overloading.

Code:-

```
package javaprogram;
public class M_overloading
{
   void add(int a, int b)
        {
                System.out.println(a+b);
         }
    void add(int a,double b)
```

```
        {
                System.out.println(a+b);
         }
  void add(double a, double b)
        {
                 System.out.println(a+b);
         }
  void add(double a, double b, double c)
        {
                System.out.println(a+b+c);
        }
  public static void main(String[] args)
        {
                M_overloading m = new M_overloading();
                m.add(7,3);
                m.add(12,2.5);
                m.add(14.2,5.8);
                m.add(2.4,8.9,6.2);
         }
}
```

Output:

```
    10

    14.5

    20.0

    17.5
```

Week-3

Aim– Code, execute and debug programs that uses different types of Constructors, Expression evaluation.

Experiment -3.1 : Write a java program to illustrate different types of Constructors.

```
package mypack;
public class Student1
{
	public String name;
	public int age;

	public Student1()
	{
		this.name="Raju";
		this.age=30;
	}
	public void display()
	{
		System.out.println("Name of the Student:"+this.name);
		System.out.println("Age of the Student:"+this.age);
	}
	public static void main(String[] args)
	{
		new Student1().display();
	}

}
```

Output:

```
	Name of the Student: Raju
	Age of the Student: 30
```

Experiment -3.2 :Write a java program for Expression Evaluation that uses different types of Operators.

Code:

```
package javaprogram;
public class Operators
{
    public static void main(String[] args)
   {
            int a = 4;
            int b = 7;
            int var;
         System.out.println("ArithematicOperators :");
         System.out.println("a + b = " + (a + b));
         System.out.println("a - b = " + (a - b));
         System.out.println("a * b = " + (a * b));
         System.out.println("a / b = " + (a / b));
         System.out.println("a % b = " + (a % b));

         System.out.println("--------------------------------------");
         System.out.println("Assignment Operators :");
         System.out.println("--------------------------------------");
            var = a;
         System.out.println("var using =: " + var);
            var += a;
         System.out.println("var using +=: " + var);
            var *= a;
         System.out.println("var using *=: " + var);

         System.out.println("--------------------------------------");
         System.out.println("Relational Operators :");
         System.out.println("--------------------------------------");
         System.out.println("a is " + a + " and b is " + b);
         System.out.println(a == b);
         System.out.println(a != b);
         System.out.println(a > b);
         System.out.println(a < b);
         System.out.println(a >= b);
         System.out.println(a <= b);

         System.out.println("--------------------------------------");
         System.out.println("Logical Operators :");
         System.out.println("--------------------------------------");
         System.out.println((5 > 3) && (8 > 5));
         System.out.println((5 > 3) && (8 < 5));
         System.out.println((5 < 3) || (8 > 5));
         System.out.println((5 > 3) || (8 < 5));
```

```
      System.out.println((5 < 3) || (8 < 5));
      System.out.println(!(5 == 3));
      System.out.println(!(5 > 3));

      System.out.println("--------------------------------------");
      System.out.println("Unary Operators :");
      System.out.println("--------------------------------------");
        int result1, result2;
      System.out.println("Value of a: " + a);
        result1 = ++a;
      System.out.println("After increment: " + result1);
      System.out.println("Value of b: " + b);
        result2 = --b;
      System.out.println("After decrement: " + result2);
    }
}
```

Output:

```
ArithematicOperators :
a + b = 11
a - b = -3
a * b = 28
a / b = 0
a % b = 4
--------------------------------------
Assignment Operators :
--------------------------------------
var using =: 4
var using +=: 8
var using *=: 32
--------------------------------------
Relational Operators :
--------------------------------------
a is 4 and b is 7
false
true
false
true
false
true
--------------------------------------
Logical Operators :
--------------------------------------
true
false
true
true
```

```
false
true
false
---------------------------------------
Unary Operators :
---------------------------------------
Value of a: 4
After increment: 5
Value of b: 7
After decrement: 6
```

Week-4

Aim: Install memory monitoring tool and observe how JVM allocates memory

Experiment-4.1: Installing Visual VM tool.

Experiment -4.1: Installation of Memory Management Tool.

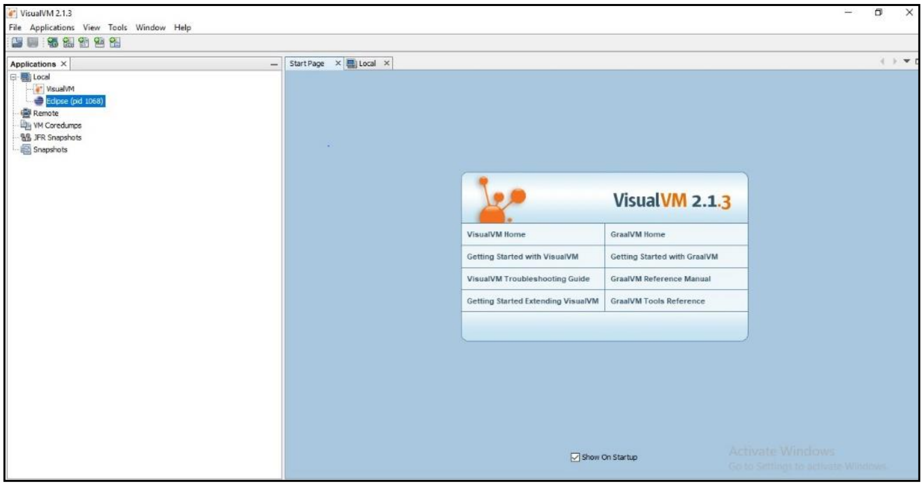
Steps:

Step-1) Open your browser and search for Visual VM

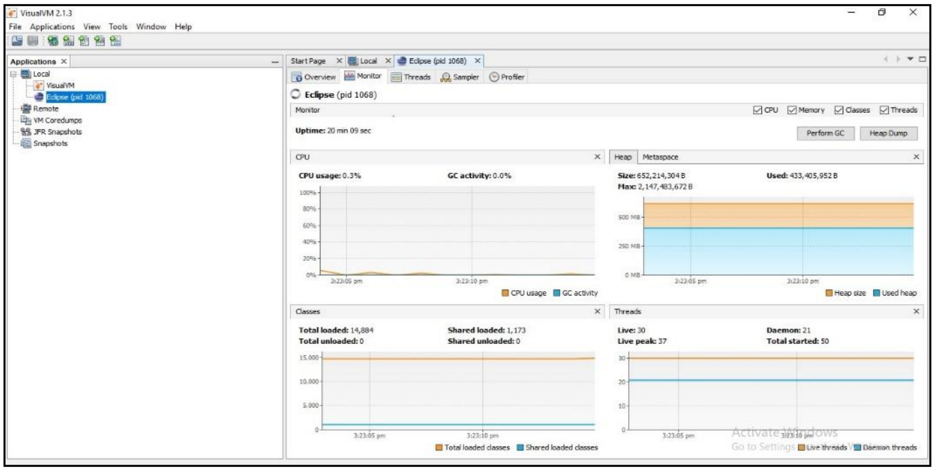
Step-2) Click on the latest version and click download

Step-3) Complete the installation procedure

Step-4) Launch the software



Step-5)After launching the software observe the changes In graph and data



Uptime: 20 min 09 sec

CPU usage: 0.3% GC activity: 0.0%

Size: 652,214,304 B Used: 433,405,952 B
Max: 2,147,483,672 B

Total loaded: 14,884 Shared loaded: 1,173
Total unloaded: 0 Shared unloaded: 0

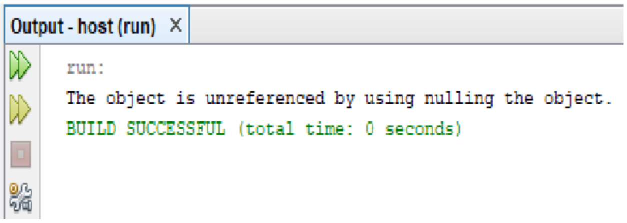
Live: 30 Daemon: 21
Live peak: 37 Total started: 50

Experiment -4.2: Memory allocation explanation through the programs.

Code -1: By nulling the reference.

```
package javaprogram;
public class Employee
{
   public void finalize()
   {
          System.out.println("The object is unreferenced by using nulling the object.");
   }
   public static void main(String[] args)
   {
      Employee E1 = new Employee();
      E1=null;
      System.gc();
   }
}
```

Output:

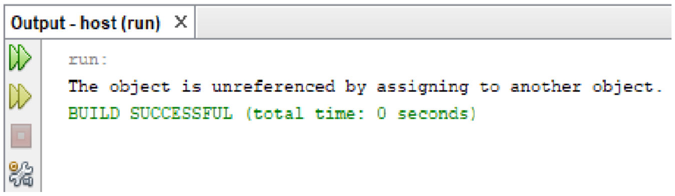


```
run:
The object is unreferenced by using nulling the object.
BUILD SUCCESSFUL (total time: 0 seconds)
```

Code -2: By assigning a reference to another object.

```
package javaprogram;
public class Employee1
{
  public void finalize()
   {
          System.out.println("The object is unreferenced by assigning to another object.");
   }
   public static void main(String[] args)
   {
      Employee1 e1 = new Employee1();
      Employee1 e2 = new Employee1();
      e1=e2;
      System.gc();
   }
}
```

Output:

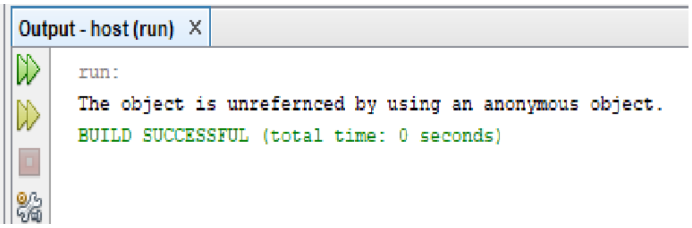


```
run:
The object is unreferenced by assigning to another object.
BUILD SUCCESSFUL (total time: 0 seconds)
```

Code-3: By creating anonymous object.

```
package javaprogram;
public class Employee2
{
    public void finalize()
   {
           System.out.println("The object is unrefernced by using an anonymous object.");
    }
   public static void main(String[] args)
   {
       new Employee2();
       System.gc();
    }
}
```

Output :



```
run:
The object is unrefernced by using an anonymous object.
BUILD SUCCESSFUL (total time: 0 seconds)
```

Week-5

Aim– Code, execute and debug programs that uses different control statements.

Experiment -5.1: Write a java program to check whether year is leap or not.

Code:

```
package mypack;
public class Leap
{
        public static void main(String[] args)
        {
                int year=2020;
                if(((year%4==0)&&(year%100!=0))||(year%400==0))
                {
                        System.out.println("LEAP YEAR");
                }
                else
                {
                        System.out.println("Common Year");
                }
        }
}
```

Output:

```
        LEAP YEAR
```

Experiment -5.2: Write a java Program to print pyramid

Code:

```
package mypack;
public class Pyramid
{
        public static void main(String[] args)
        {
                int row=5;
                for(int i=1; i<=row; i++)
                {
                        for(int j=1; j<=i; j++)
                        {
                                System.out.print("*");
                        }
                        System.out.println();
                }
        }
}
```

Output:

```
*
**
***
****
*****
```

Experiment -5.3 Write a java Program to calculate sum of first 20 natural numbers

Code:

```
package mypack;
public class NaturalNum
{
	public static void main(String[] args)
	{
		int sum=0;
		for(int i=1; i<=20; i++)
		{
			sum=sum+i;
		}
		System.out.println("The Sum of natural number is "+sum);
	}

}
```

Output:

```
The Sum of natural number is 210
```

Experiment -5.4 Write a java Program to check given number is palindrome or not

Code:

```
package mypack;
public class Palindrome
{
	public static void main(String[] args)
	{
		int r, sum=0, temp;
		int n=454;
		temp=n;
		while(n>0)
		{
			r=n%10;
			sum=(sum*10)+r;
			n=n/10;
		}

		if(temp==sum)
```

```
                        System.out.println("Palindrome Number");
                else
                        System.out.println("Not Palindrome");
        }
}
```

Output:

Palindrome Number

Experiment -5.5 Write a java Program to check given number is Prime or not

Code:

```
package mypack;
public class Prime
{
        public static void main(String[] args)
        {
                int i, m=0, flag=0;
                int n=4;
                m=n/2;
                if(n==0||n==1)
                {
                        System.out.println(n+ " is not Prime Number");
                }
                else
                {
                        for(i=2; i<=m; i++)
                        {
                                if(n%i==0)
                                {
                                        System.out.println(n+ " is not Prime Number");
                                        flag =1;
                                        break;
                                }
                        }
                        if(flag==0)
                        {
                                System.out.println(n + "  Prime Number");
                        }
                }
        }
}
```

Output:

```
4
Is not Prime Number
```

Experiment -5.6 Write a java Program to check whether alphabets is vowel or consonant.

Code:

```
package mypack;
public class Switchvowel
{
        public static void main(String[] args)
        {
                char ch ='A';
                switch(ch)
                {
                        case 'a':
                                System.out.println("Vowel");
                                break;
                        case 'e':
                                System.out.println("Vowel");
                                break;
                        case 'i':
                                System.out.println("Vowel");
                                break;
                        case 'o':
                                System.out.println("Vowel");
                                break;
                        case 'u':
                                System.out.println("Vowel");
                                break;
                        case 'A':
                                System.out.println("Vowel");
                                break;
                        case 'E':
                                System.out.println("Vowel");
                                break;
                        case 'I':
                                System.out.println("Vowel");
                                break;
                        case 'O':
                                System.out.println("Vowel");
                                break;
                        case 'U':
                                System.out.println("Vowel");
                                break;
                        default:
                                System.out.println("Consonant");
                }
        }
}
```

Output:

```
Vowel
```

Experiment -5.7 Write a java Program to check whether the given number is Positive, Negative or Zero.

Code:

```
package mypack;
public class Positive
{
	public static void main(String[] args)
	{
		int number = -13;
		if(number > 0)
		{
			System.out.println("POSITIVE");
		}
		else if(number < 0)
		{
			System.out.println("NEGATIVE");
		}
		else
		{
			System.out.println("ZERO");
		}
	}
}
```

Output:

```
	NEGATIVE
```

Week-6

Aim–Code, execute and debug programs that uses encapsulation.

Experiment -6.1: Write a java program to print Person details using Encapsulation concept.

Class 1:

```
package mypack;
public class encap_person
{
   private int age;
   private String name;
   public void setdata(int age, String name)
   {
          this.age=age;
          this.name=name;
   }
   public void getdata()
   {
          System.out.println("The name of person is :" +name+"\n"+"The age of person is : "+ age);
   }

}
```

Class 2:

```
package mypack;
public class person_test
{
          public static void main(String[] args)
          {
                    encap_person p = new encap_person();
                    p.setdata(20,"Prasad");
                    p.getdata();
          }
}
```

Output:

```
The name of person is :Prasad
The age of person is : 20
```

Experiment - 6.2: Define class and implement simple calculator using SRP.

Class 1:

```
package mypack;
public class addition
{
   private int num1;
   private int num2;
   public void setdata(int num1, int num2)
   {
      this.num1=num1;
      this.num2=num2;
   }
   public void getdata()
   {
      int result=(num1+num2);
      System.out.println(result);
   }
}
```

Class 2:

```
package mypack;
public class substraction
{
   private int num1;
   private int num2;
   public void setdata(int num1,int num2)
   {
      this.num1=num1;
      this.num2=num2;
   }
   public void getdata()
   {
      int result=(num1-num2);
      System.out.println(result);
   }

}
```

Class 3:

```
package mypack;
public class multiplication
{
         private int num1;
   private int num2;
   public void setdata(int num1,int num2)
   {
      this.num1=num1;
      this.num2=num2;
   }
   public void getdata()
   {
      int result=(num1*num2);
      System.out.println(result);
   }
}
```

Class 4:

```
package mypack;
public class division
{
          private int num1;
   private int num2;
   public void setdata(int num1,int num2)
   {
      this.num1=num1;
      this.num2=num2;
   }
   public void getdata()
   {
      int result=(num1/num2);
      System.out.println(result);
   }
}
```

Class 5:

```
package mypack;
import java.util.Scanner;
public class Calculator
{
          public static void main(String[] args)
          {
                    while (true)
                    {
                              Scanner s=new Scanner(System.in);

                              addition a=new addition();
                              substraction sub=new substraction();
                              multiplication m=new multiplication();
                              division d=new division();

                              System.out.println("Select from the following choices-");
                              System.out.println("1 for Addition");
                              System.out.println("2 for Subtraction");
                              System.out.println("3 for Multiplication");
                              System.out.println("4 for Division");
                              System.out.println("Enter your choice:");

                              int choice = s.nextInt();
                              if (choice >= 1 && choice <= 4)
                    {
                              System.out.println("Enter the first number:");
                              int num1 = s.nextInt();
                              System.out.println("Enter the second number:");
                              int num2 = s.nextInt();
                              System.out.println("------------------");

                           switch (choice)
                               {
                                 case 1:
                                      a.setdata(num1, num2);
                                      a.getdata();
                                      break;
                                case 2:
                                      sub.setdata(num1,num2);
                                      sub.getdata();
                                      break;
                               case 3:
                                      m.setdata(num1, num2);
                                      m.getdata();
                                      break;
                              default:
                                      d.setdata(num1, num2);
                                      d.getdata();

                               }
                    }
                }
          }
}
```

Output:

```
Select from the following choices-
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
Enter your choice:
1
Enter the first number:
11
Enter the second number:
22
------------------
33
```

```
Select from the following choices-
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
Enter your choice
2
Enter the first number:
55
Enter the second number:
66
------------------
-11

Select from the following choices-
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
Enter your choice:
3
Enter the first number:
5
Enter the second number:
5
------------------
25

Select from the following choices-
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
Enter your choice:
4
Enter the first number:
10
Enter the second number:
5
------------------
2

Select from the following choices-
1 for Addition
2 for Subtraction
3 for Multiplication
4 for Division
Enter your choice:
```

Week -7

Aim–Code, execute and debug programs that uses array concepts.

Experiment -7.1: Write a java program to find sum and average of an array.

Code:

```
package mypack;
public class array_demo
{
   public static void main(String[] args)
   {
             int [] a={1,2,3,4,5,6,7};
            int sum=0;
            for(int i=0; i<a.length; i++)
            {
                        sum=sum+a[i];
            }
            System.out.println("Sum :"+sum);
            System.out.println("Average :"+sum/a.length);
   }
}
```

Output:

```
Sum :28
Average :4
```

Experiment -7.2: Write a java program to add two matrices using array.

Code :

```
package mypack;
public class mul
{
   public static void main(String[] args)
   {
            int[][] a={{2,3},{4,5}};
            int[][] b={{3,6},{4,1}};
            int[][] c= new int[2][2];
            for (int i=0; i<2; i++)
            {
                      for (int j=0; j<2; j++)
                      {
                                c[i][j]=a[i][j]+b[i][j];
                                System.out.print(c[i][j]+"  ");
                      }
                      System.out.println();
             }
   }
}
```

Output:

```
5  9
8  6
```

Experiment -7.3: Code, execute and debug programs to perform string manipulation (String methods).

Code:

```
package mypack;

public class Str_methods

{

        public static void main(String[] args)

        {

                String s1="Java Program";

                String s2="Jain Poltytechnic Belagavi";

               System.out.println(s1.length());

               System.out.println(s1.charAt(2));

               System.out.println(s1.indexOf('P'));

               System.out.println(s1.toLowerCase());

               System.out.println(s1.toUpperCase());

               System.out.println(s1.concat(s2));

               System.out.println(s1.substring(2));

               System.out.println(s1.equals(s2));

               System.out.println(s2=s1.replace('J', 'd'));

               System.out.println(s1.contains(s2));

         }

}
```

Output:

```
12
v
5
java program
JAVA PROGRAM
Java ProgramJain Poltytechnic Belagavi
va Program
false
dava Program
false
```

Experiment -7.4: Write a java program to check whether given String is Palindrome or Not.

Code:

```
package mypack;
public class Str_palindrome
{
	public static void main(String[] args)
	{
		String string ="MALAYALAM";
		Boolean flag= true;
		for(int i=0; i<string.length( ) / 2; i++)
		{
			if(string.charAt(i)!=string.charAt(string.length()-i-1))
			{
				flag= false;
				break;
			}
		}
		if (flag)
			System.out.println("Palindrome");
		else
			System.out.println("Not Palindrome");

	}
}
```

Output:

```
	Palindrome
```

Week–8

Aim: Code, execute and debug java programs that uses inheritance concept.

Experiment -8.1: Write a java program for Single level Inheritance.

Code:

```
package mypack;
class one
{
	public void print_one()
	{
		System.out.println("Java");
	}
}
class two extends one
{
	public void print_two()
	{
		System.out.println("Program");
	}
}
public class main
{
	public static void main(String[] args)
	{
		two t=new two();
		t.print_one();
		t.print_two();
	}
}
```

Output:

```
	Java
	Program
```

Experiment -8.2: Write a java program for Multi-Level Inheritance.

Code:

```
package mypack;
class one
{
        public void print_one()
        {
                System.out.println("Java");
        }
}
class two extends one
{
        public void print_two()
        {
                System.out.println("Program");
        }
}
class three extends two
{
        public void print_three()
        {
                System.out.println("Examples");
        }
}
public class multilevel
{
        public static void main(String[] args)
        {
                three t= new three();
                t.print_one();
                t.print_two();
                t.print_three();
        }
}
```

Output:

```
        Java
        Program
        Examples
```

Experiment -8.3: Write a java program for Hierarchical Inheritance.

Code:

```
package mypack;
class A
{
        public void print_A()
        {
                System.out.println("This is Class A");
        }
}
class B extends A
{
        public void print_B()
        {
                System.out.println("This is class B");
        }
}
class C extends A
{
        public void print_C()
        {
                System.out.println("This is Class C");
        }
}
class D extends A
{
        public void print_D()
        {
                System.out.println("This is Class D");
        }
}
public class Hierarchical
{
        public static void main(String[] args)
        {
                B obj_B= new B();
                obj_B.print_A();
                obj_B.print_B();

                C obj_C= new C();
                obj_C.print_A();
                obj_C.print_C();

                D obj_D= new D();
                obj_D.print_A();
                obj_D.print_D();
        }
}
```

Output:

```
        This is Class A
        This is class B
        This is Class A
        This is Class C
        This is Class A
        This is Class D
```

Experiment - 8.4: Write a java program to find area of Rectangle and Circle

Code:

```
package mypack;
interface shape
{
        public double calculateArea();
}
class Rect implements shape
{
        double length;
        double width;
        public double calculateArea()
        {
                return length*width;
        }
}
class circle implements shape
{
        public double radius;
        public double calculateArea()
        {
                return(22.0/7)*radius*radius;
        }
}
class Area
{
        public double calculateShapeArea(shape shape)
        {
                return shape.calculateArea();
        }
}
public class Inheritance
{
        public static void main(String[] args)
        {
                Rect r= new Rect();
                r.length=10.5;
                r.width=20.0;
                Area a= new Area();
                System.out.println(a.calculateShapeArea(r));

                circle c= new circle();
                c.radius=1.0;
                System.out.println(a.calculateShapeArea(c));

        }
}
```

Output:

```
        210.0
        3.142857142857143
```

Week – 9

Aim: Code, execute and debug programs that uses Static binding/Compile time polymorphism and Dynamic binding/Runtime polymorphism.

Experiment -9.1: Write a program to achieve static binding/compile time polymorphism.

Code:

```
package mypack;
public class Compile
{
        void add(int a, int b)
        {
            System.out.println(a+b);
        }
        void add(int a,double b)
        {
            System.out.println(a+b);
        }
        void add(double a, double b)
        {
            System.out.println(a+b);
        }
        void add(double a, double b, double c)
        {
            System.out.println(a+b+c);
        }
public static void main(String[] args)
      {
            Compile c= new Compile();
            c.add(5,7);
            c.add(5,2.5);
            c.add(7.2,3.8);
            c.add(5.4,1.9,7.2);
      }
}
```

Output:

```
12
7.5
11.0
14.5
```

Experiment -9.2: Write a program to achieve dynamic binding/runtime polymorphism.

Code:

```
package mypack;
class Person
{
	public void Speak()
	{
		System.out.println("Person Speak");
	}
}
class Teacher extends Person
{
	public void Speak()
	{
		System.out.println("Teacher Speak");
	}
}
public class Runtime
{
	public static void main(String[] args)
	{
		Person obj1= new Person();
		obj1.Speak();
		Person obj =new Teacher();
		obj.Speak();
	}
}
```

Output:

```
	Person Speak
	Teacher Speak
```

Week - 10

Aim: Code, execute and debug program that uses abstraction concept.

Experiment -10.1: Write a java program to achieve abstraction using abstract class.

Class 1:

```
package mypack;
abstract class Person
{
   abstract void work();
   abstract void eat();
}
```

Class 2:

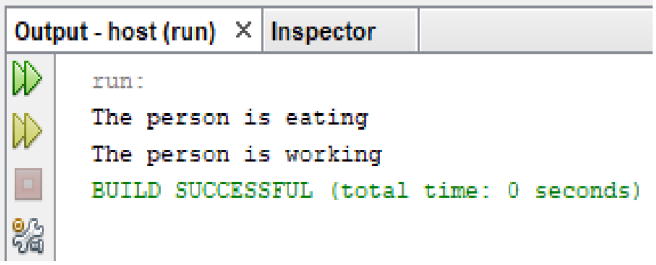
```
package mypack;
public class Employee extends Person
{
   public void work()
   {
          System.out.println("The person is working");
   }
   public void eat()
   {
          System.out.println("The person is eating");
    }
}
```

Class 3:

```
Package mypack;
public class abs_test {
   public static void main(String[] args)
   {
          Person p = new Employee();
          p.eat();
```

```
            p.work();
    }
}
```

Output:



```
run:
The person is eating
The person is working
BUILD SUCCESSFUL (total time: 0 seconds)
```

Experiment -10.2: Write a java program to achieve abstraction using interface.

Class 1:

```
package mypack;
public interface college
{
    abstract void enjoy();
}
```

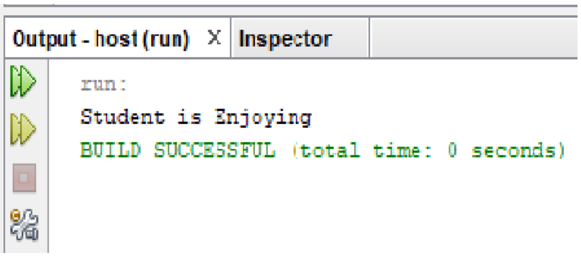
Class 2:

```
package mypack;
public class student implements college
{
            public void enjoy()
            {
                        System.out.println("Student are Enjoying");
             }
}
```

Class 3:

```
package mypack;
public class Int_test
{
    public static void main(String[] args)
    {
            college c = new student();
            c.enjoy();
    }
}
```

Output:



```
run:
Student is Enjoying
BUILD SUCCESSFUL (total time: 0 seconds)
```

Experiment -10.3: Write a java program to achieve multiple inheritance using interface.

Class 1:

```
package Abstraction;
interface A
{
    void draw();
}
```

Class 2:

```
package Abstraction;
interface B
{
    void area();
}
```

Class 3:

```
package Abstraction;
 class C implements A,B
{
        public void draw()
        {
                System.out.println("Square set is 4cm");
                System.out.println("Rectangle has 4 sides");
         }
        public void area()
         {
                 int a=4;
                int b=6;
                System.out.println("Area of square :"+a*a);
                System.out.println("Area of Rectangle :"+a*b);
          }
}
```

Class 4:

```
package Abstraction;
public class IMI_test
{
   public static void main(String[] args)
   {
      C c1 = new C();
      c1.draw();
      c1.area();
   }
}
```

Output:

```
        Square set is 4cm
        Rectangle has 4 sides
        Area of square: 16
        Area of Rectangle: 24
```

Week 11:

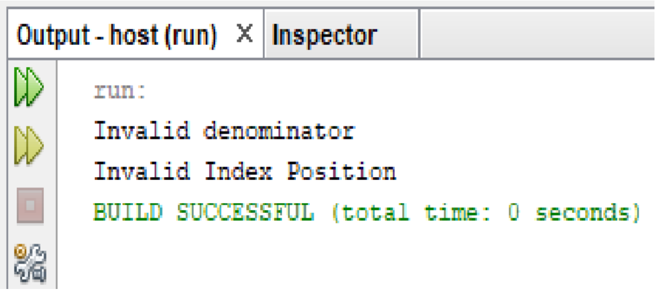
Aim: Code, execute and debug programs in java to achieve file handling and exception handling.

Experiment 11.1: Write a java program for Unchecked Exception.

Code : (Arithmetic Exception)

```
package mypack;
public class ArithException
{
   public static void main(String[] args)
   {
         try
         {
                  System.out.println(10/0);
         }
         catch(ArithmeticException e)
         {
                  System.out.println("Invalid denominator");
         }
      /*catch(ArrayIndexOutOfBoundException e)*/
         {
                  System.out.println("Invalid Index Position");
         }
   }
}
```

Output:

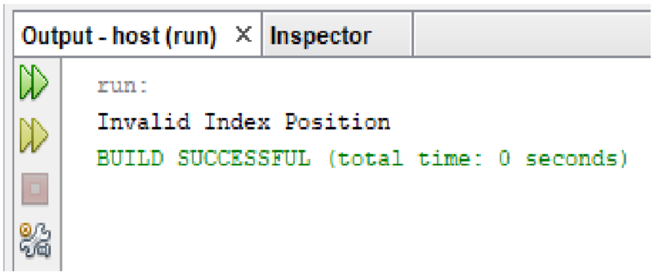


```
run:
Invalid denominator
Invalid Index Position
BUILD SUCCESSFUL (total time: 0 seconds)
```

Code: (Array Exception)

```
package  mypack;
public class ArrException
 {
   public static void main(String[] args)
   {
       try
       {
           int[] arr={1,2,3};
           System.out.println(arr[10]);
       }
      catch(ArrayIndexOutOfBoundsException e)
       {
           System.out.println("Invalid Index Position");
       }
   }
}
```

Output:



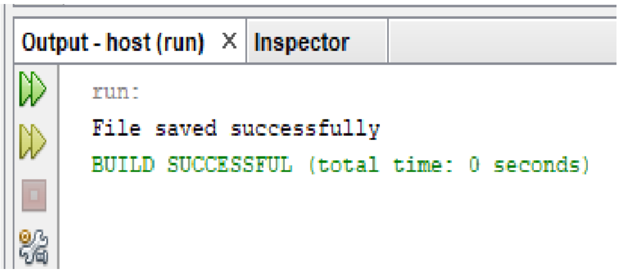
```
run:
Invalid Index Position
BUILD SUCCESSFUL (total time: 0 seconds)
```

Experiment -11.2 :Write a java program for Checked Exception.

Code:(FileNotFoundException)

```
package  mypack;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
public class FNFE
 {
   public static void main(String[] args)
   {
            PrintWriter pw;
             try
             {
                          pw=new PrintWriter("ftp.txt");
                          pw.println("Saved");
            }
            catch(FileNotFoundException e)
            {
                          System.out.println(e);
            }
            System.out.println("File saved successfully");
   }
}
```

Output:



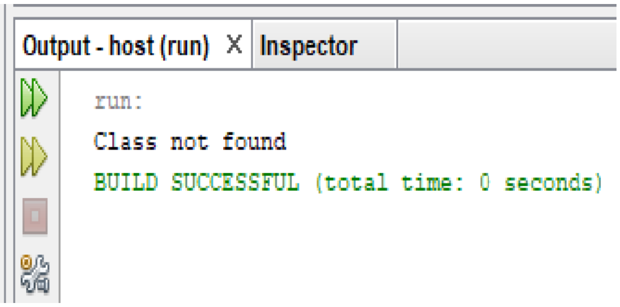
```
run:
File saved successfully
BUILD SUCCESSFUL (total time: 0 seconds)
```

Code: (ClassNotFoundException)

```
package Exception;
public class check_demo
 {
   public static void main(String[] args)
   {
       try
```

```
        {
            Class.forName("NOClassExist");
        }
            catch(ClassNotFoundException e)
        {
            System.out.println("Class not found");
        }
    }
}
```

Output:



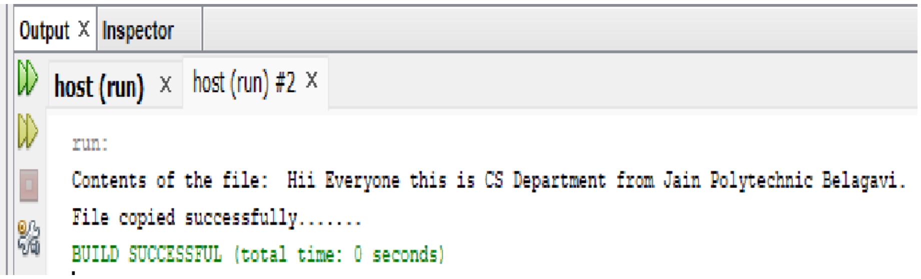
```
run:
Class not found
BUILD SUCCESSFUL (total time: 0 seconds)
```

Experiment -11.3: Write a java program that reads the content of a file and write the content to another file.

```
package mypack;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;
public class CopyContent
{
    public static void main(String[] args) throws IOException
        {
            File file = new File("C:\\Users\\Lenovo\\Documents\\a.txt");
```

```
                    FileInputStreaminputStream = new FileInputStream(file);
                    Scanner sc = new Scanner(inputStream);
                    StringBuffer buffer = new StringBuffer();
                    while(sc.hasNext())
                    {
                              buffer.append(" "+sc.nextLine());
                    }
                    System.out.println("Contents of the file: "+buffer);
                    File dest = new File("C:\\Users\\Lenovo\\Documents\\b.txt");
                    FileWriter writer = new FileWriter(dest);
                    writer.write(buffer.toString());
                    writer.flush();
                    System.out.println("File copied successfully.......");
          }
}
```

Output:



```
run:
Contents of the file:  Hii Everyone this is CS Department from Jain Polytechnic Belagavi.
File copied successfully.......
BUILD SUCCESSFUL (total time: 0 seconds)
```

Week – 12

Aim :Design an interface and implement it like one that builds different types of toys and check compliance with ISP.

Code :

Class 1:

```
package Interface;
public interface Toy
{
   void setprice(double price);
   void setcolor(String color);
}
```

Class 2:

```
package Interface;
public interface flyable
{
   void fly();
}
```

Class 3:

```
package Interface;
public interface movable
{
   void start();
   void stop();
 }
```

Class4:

```
package Interface;
public class Car implements Toy,movable
{
   double price;
   String color;
```

```
 @Override
 public void setprice(double price)
 {
        this.price=price;
 }
 @Override
 public void setcolor(String color)
 {
        this.color=color;
 }
 @Override
 public void start()
 {
        System.out.println("The car is started");
 }
 @Override
 public void stop()
 {
        System.out.println("The car has stopped");
 }
}
```

Class5:

```
package Interface;
public class Aeroplane implements Toy,movable,flyable
{
 double price;
 String color;
@Override
public void setprice(double price)
{
```

```
this.price=price;
}
@Override
public void setcolor(String color)
{
this.color=color;
}
@Override
public void start()
{
System.out.println("The Aeroplane is started");
}
@Override
public void stop()
{
System.out.println("The Aeroplane has stopped");
}
@Override
public void fly()
{
System.out.println("The Aeroplane is flying");
}
}
```

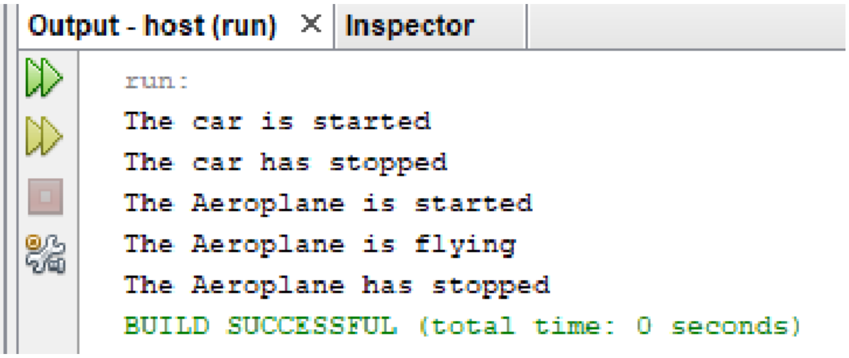
Class6:

```
package Interface;
public class Toy_test
{
  public static void main(String[] args)
  {
        Car c = new Car();
```

```
            c.start();
            c.stop();
            c.setcolor("Black");
            c.setprice(100000);

            Aeroplane a = new Aeroplane();
            a.start();
            a.fly();
            a.stop();
            a.setcolor("Blue");
            a.setprice(100000000);
    }
}
```

Output:



```
run:
The car is started
The car has stopped
The Aeroplane is started
The Aeroplane is flying
The Aeroplane has stopped
BUILD SUCCESSFUL (total time: 0 seconds)
```