Introduction to Java:

JAVA was developed by **James Gosling** at **Sun Microsystems** in the year **1991**, later acquired by Oracle Corporation. It is a simple programming language. Java makes writing, compiling, and debugging programming easy. It helps to create reusable code and modular programs.

Java is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible. A general-purpose programming language made for developers to write once run anywhere that is compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can run on any Java Virtual Machine. The syntax of Java is similar to C/C++.

History

Java's history is very interesting. It is a programming language created in 1991. James Gosling, Mike Sheridan, and Patrick Naughton, a team of Sun engineers known as the **Green team** initiated the Java language in 1991. **Sun Microsystems** released its first public implementation in 1996 as **Java 1.0**.

It provides no-cost -run-times on popular platforms. Java1.0 compiler was re-written in Java by Arthur Van Hoff to strictly comply with its specifications. With the arrival of Java 2, new versions had multiple configurations built for different types of platforms.

In 1997, Sun Microsystems approached the ISO standards body and later formalized Java, but it soon withdrew from the process. At one time, Sun made most of its Java implementations available without charge, despite their proprietary software status. Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System.

On November 13, 2006, Sun released much of its Java virtual machine as free, open-source software. On May 8, 2007, Sun finished the process, making all of its JVM's core code available under open-source distribution terms.

The principles for creating java were simple, robust, secured, high performance, portable, multi-threaded, interpreted, dynamic, etc. **James Gosling in 1995** developed Java, who is known as the **Father of Java**. Currently, Java is used in mobile devices, internet programming, games, e-business, etc.

Java programming language is named JAVA. Why?

After the name OAK, the team decided to give a new name to it and the suggested words were Silk, Jolt, revolutionary, DNA, dynamic, etc. These all names were easy to spell and fun to say, but they all wanted the name to reflect the essence of technology. In accordance with James Gosling, **Java** the among the top names along with **Silk**, and since java was a unique name so most of them preferred it.

Java is the name of an **island** in Indonesia where the first coffee(named java coffee) was produced. And this name was chosen by James Gosling while having coffee near his office. Note that Java is just a name, not an acronym.

Java Terminology

Before learning Java, one must be familiar with these common terms of Java.

1. Java Virtual Machine(JVM): This is generally referred to as JVM. There are three execution phases of a program. They are written, compile and run the program.

- Writing a program is done by a java programmer like you and me.
- The compilation is done by the **JAVAC** compiler which is a primary Java compiler included in the Java development kit (JDK). It takes Java program as input and generates bytecode as output.
- In the Running phase of a program, **JVM** executes the bytecode generated by the compiler.
- Now, we understood that the function of Java Virtual Machine is to execute the bytecode produced by the compiler. Every Operating System has a different JVM but the output they produce after the execution of bytecode is the same across all the operating systems. This is why Java is known as a **platform-independent language.**

2. Bytecode in the **Development process:** As discussed, the Javac compiler of JDK compiles the java source code into bytecode so that it can be executed by JVM. It is saved as **.class** file by the compiler. To view the bytecode, a disassembler like javap can be used.

3. Java Development Kit(JDK): While we were using the term JDK, when we learn about bytecode and JVM . So, as the name suggests, it is a complete Java development kit that includes everything including compiler, Java Runtime Environment (JRE), java debuggers, java docs, etc. For the program to execute in java, we need to install JDK on our computer in order to create, compile and run the java program.

4. Java Runtime Environment (JRE): JDK includes JRE. JRE installation on our computers allows the java program to run, however, we cannot compile it. JRE includes a browser, JVM, applet supports, and plugins. For running the java program, a computer needs JRE.

5. Garbage Collector: In Java, programmers can’t delete the objects. To delete or recollect that memory JVM has a program called Garbage Collector. Garbage Collectors can recollect the of objects that are not referenced. So Java makes the life of a programmer easy by handling memory management. However, programmers should be careful about their code whether they are using objects that have been used for a long time. Because Garbage cannot recover the memory of objects being referenced.

6. ClassPath: The classpath is the file path where the java runtime and Java compiler look for **.class** files to load. By default, JDK provides many libraries. If you want to include external libraries they should be added to the classpath.

Primary/Main Features of Java

1. Platform Independent: Compiler converts source code to bytecode and then the JVM executes the bytecode generated by the compiler. This bytecode can run on any platform be it Windows, Linux, macOS which means if we compile a program on Windows, then we can run it on Linux and vice versa. Each operating system has a different JVM, but the output produced by all the OS is the same after the execution of bytecode. That is why we call java a platform-independent language.

2. Object-Oriented Programming Language: Organizing the program in the terms of collection of objects is a way of object-oriented programming, each of which represents an instance of the class. The four main concepts of Object-Oriented programming are:

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

3. Simple: Java is one of the simple languages as it does not have complex features like pointers, operator overloading, multiple inheritances, Explicit memory allocation.

4. Robust: Java language is robust that means reliable. It is developed in such a way that it puts a lot of effort into checking errors as early as possible, that is why the java compiler is able to detect even those errors that are not easy to detect by another programming language. The main features of java that make it robust are garbage collection, Exception Handling, and memory allocation.

5. Secure: In java, we don't have pointers, and so we cannot access out-of-bound arrays i.e it shows **ArrayIndexOutOfBound Exception** if we try to do so. That's why several security flaws like stack corruption or buffer overflow is impossible to exploit in Java.

6. Distributed: We can create distributed applications using the java programming language. Remote Method Invocation and Enterprise Java Beans are used for creating distributed applications in java. The java programs can be easily distributed on one or more systems that are connected to each other through an internet connection.

7. Multithreading: Java supports multithreading. It is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU.

8. Portable: As we know, java code written on one machine can be run on another machine. The platform-independent feature of java in which its platform-independent bytecode can be taken to any platform for execution makes java portable.

9. High Performance: Java architecture is defined in such a way that it reduces overhead during the runtime and at some time java uses Just In Time (JIT) compiler where the compiler compiles code on-demand basics where it only compiles those methods that are called making applications to execute faster.

10. Dynamic flexibility: Java being completely object-oriented gives us the flexibility to add classes, new methods to existing classes and even creating new classes through sub-classes. Java even supports functions written in other languages such as C, C++ which are referred to as native methods.

11. Sandbox Execution: Java programs run in a separate space that allows user to execute their applications without affecting the underlying system with help of a bytecode verifier. Bytecode verifier also provides additional security as it's role is to check the code for any violation access.

12. Write Once Run Anywhere: As discussed above java application generates '.class' file which corresponds to our applications(program) but contains code in binary format. It provides ease t architecture-neutral ease as bytecode is not dependent on any machine architecture. It is the primary reason java is used in the enterprising IT industry globally worldwide.

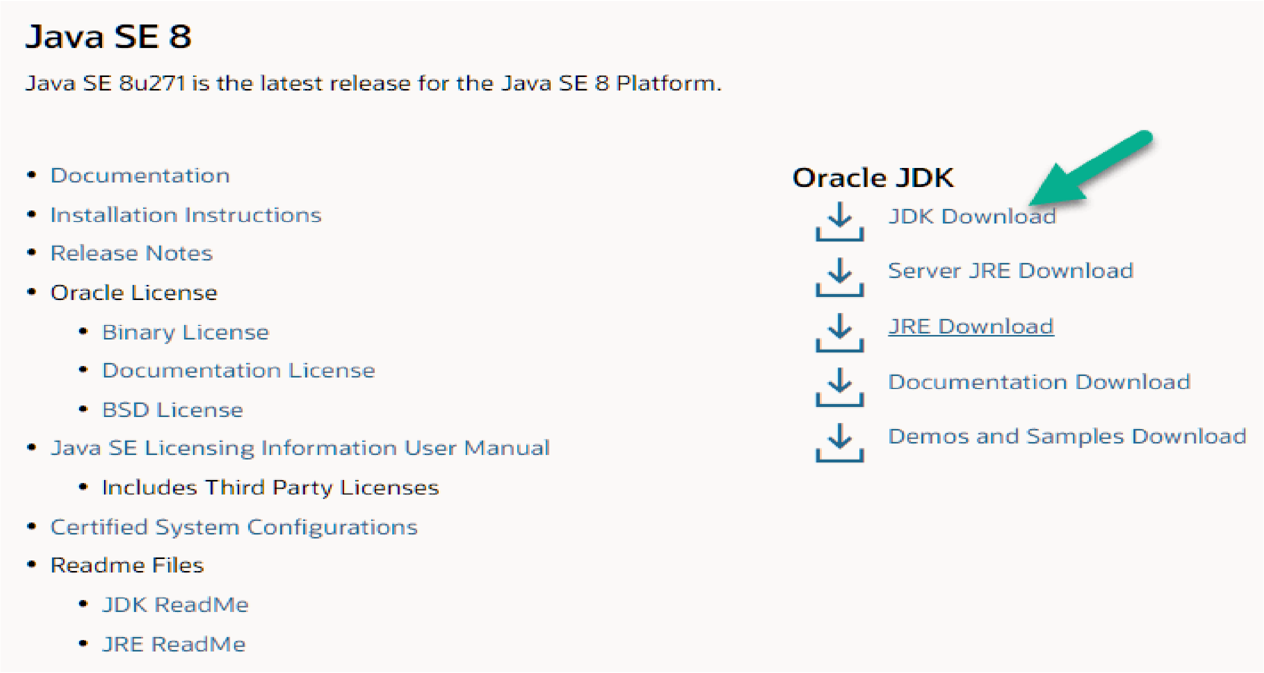
13. Power of compilation and interpretation: Most languages are designed with purpose either they are compiled language or they are interpreted language. But java integrates arising enormous power as Java compiler compiles the source code to bytecode and JVM executes this bytecode to machine OS-dependent executable code.

How to install Java for Windows

Java Development Kit(JDK) allows you to code and run Java programs. It's possible that you install multiple JDK versions on the same PC. But Its recommended installing Java on Windows 10 with latest version.

Following are the steps on how to install Java in Windows 10 for JDK 8 free download for 32 bit or JDK8 download for Windows 64 bit and installation.

Step 1) Go to www.java.com. Click on JDK Download for Java download JDK 8.



Java SE 8

Java SE 8u271 is the latest release for the Java SE 8 Platform.

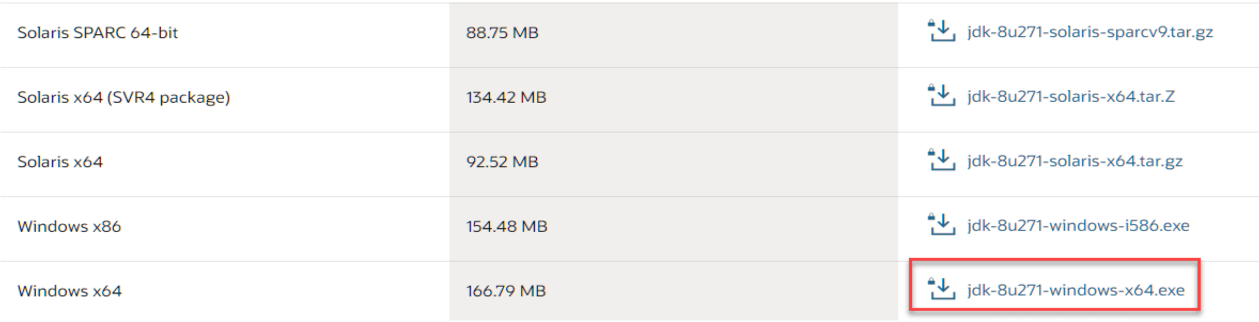
- Documentation
- Installation Instructions
- Release Notes
- Oracle License
  - Binary License
  - Documentation License
  - BSD License
- Java SE Licensing Information User Manual
  - Includes Third Party Licenses
- Certified System Configurations
- Readme Files
  - JDK ReadMe
  - JRE ReadMe

Oracle JDK

- JDK Download
- Server JRE Download
- JRE Download
- Documentation Download
- Demos and Samples Download

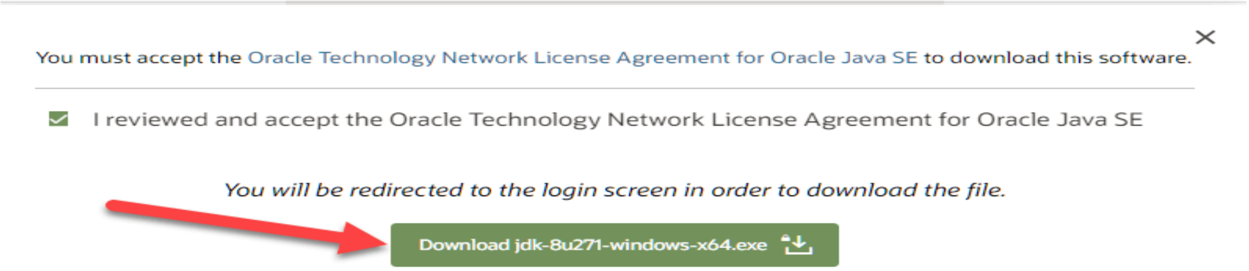
Step 2) Next,

1. Accept License Agreement
2. Download Java 8 JDK for your version 32 bit or JDK download 64 bit.



Solaris SPARC 64-bit	88.75 MB	jdk-8u271-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	134.42 MB	jdk-8u271-solaris-x64.tar.Z
Solaris x64	92.52 MB	jdk-8u271-solaris-x64.tar.gz
Windows x86	154.48 MB	jdk-8u271-windows-i586.exe
Windows x64	166.79 MB	jdk-8u271-windows-x64.exe

Step 3) When you click on the Installation link the popup will be open. Click on I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE development kit and you will be redirected to the login page. If you don't have an oracle account you can easily sign up by adding basics details of yours.



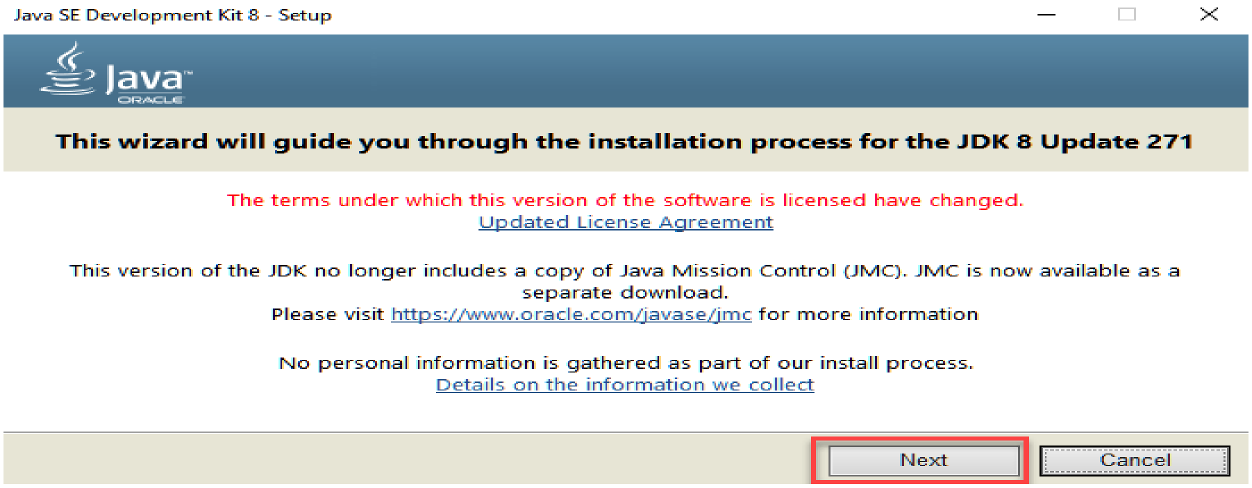
You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.

☑ I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE

You will be redirected to the login screen in order to download the file.

NOTE: You will be required to create an Oracle Account to start Java 8 download of the file.

Step 4) Once the Java JDK 8 download is complete, run the exe for install JDK. Click Next



Java SE Development Kit 8 - Setup

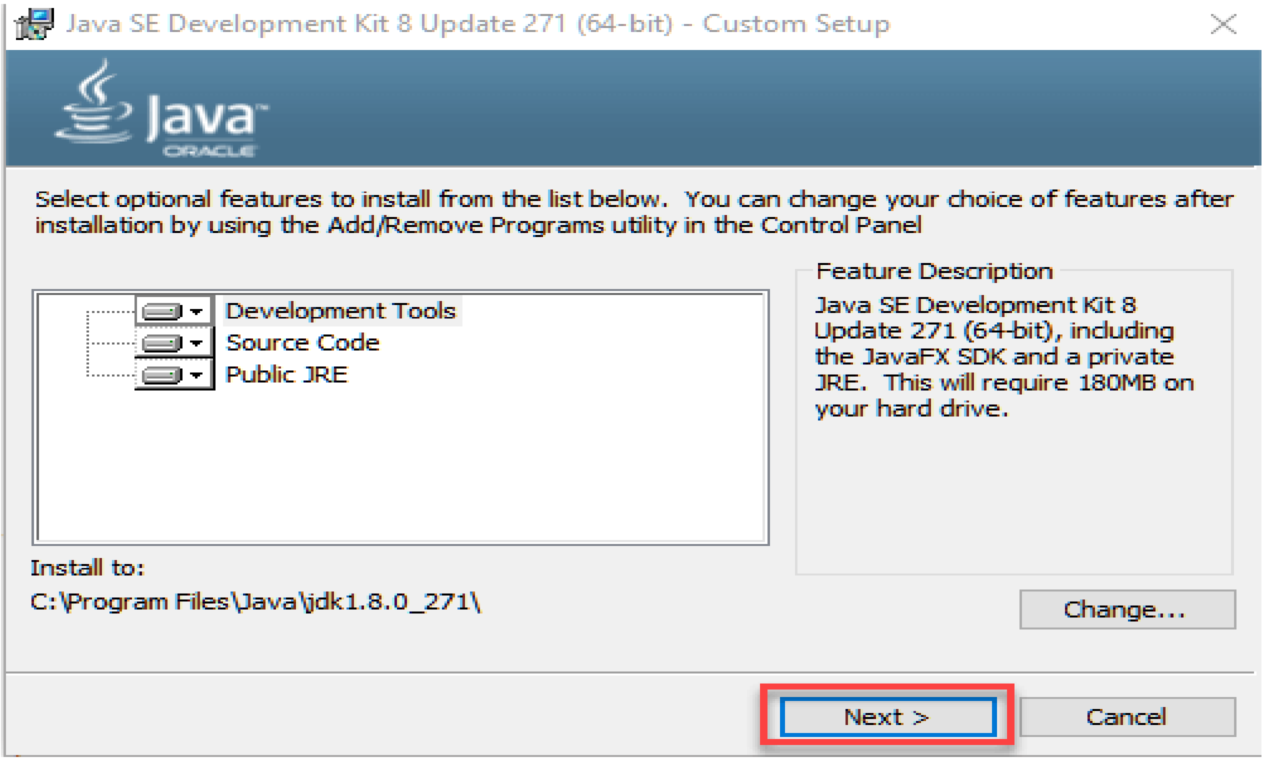
This wizard will guide you through the installation process for the JDK 8 Update 271

The terms under which this version of the software is licensed have changed.
Updated License Agreement

This version of the JDK no longer includes a copy of Java Mission Control (JMC). JMC is now available as a separate download.
Please visit https://www.oracle.com/javase/jmc for more information

No personal information is gathered as part of our install process.
Details on the information we collect

Step 5) Select the PATH to install Java in Windows… You can leave it Default. Click next.



Java SE Development Kit 8 Update 271 (64-bit) - Custom Setup
Java
ORACLE

Select optional features to install from the list below. You can change your choice of features after installation by using the Add/Remove Programs utility in the Control Panel

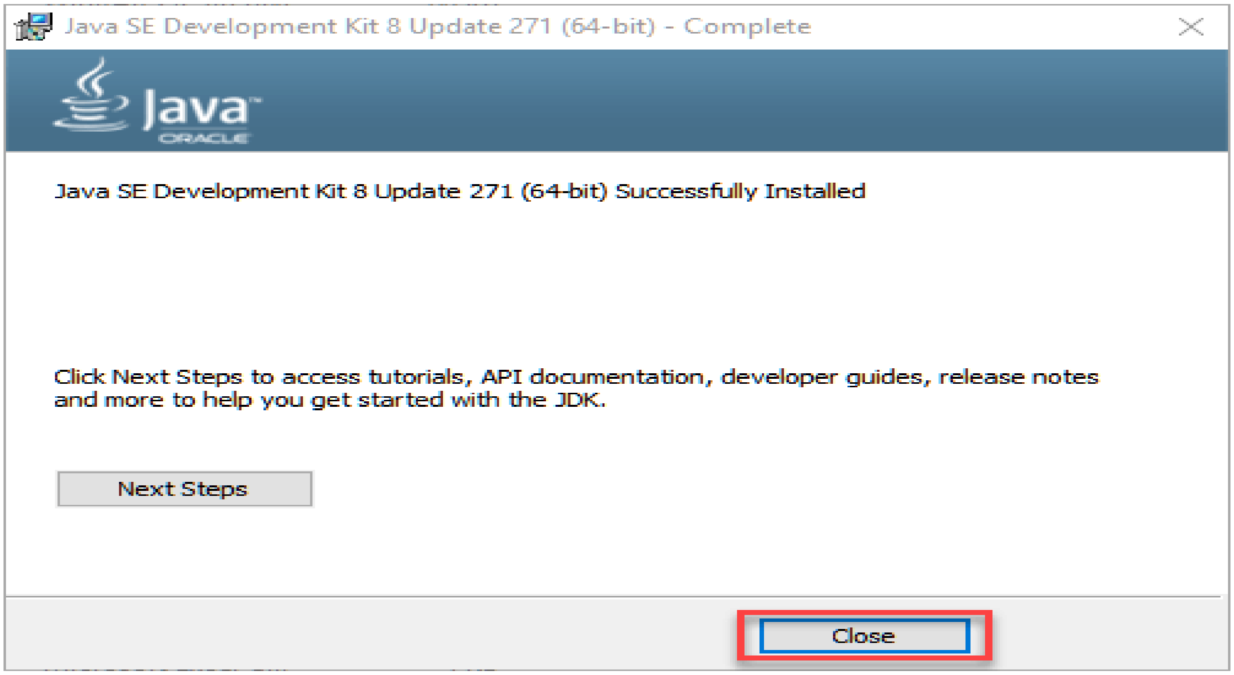
Development Tools
Source Code
Public JRE

Feature Description

Java SE Development Kit 8 Update 271 (64-bit), including the JavaFX SDK and a private JRE. This will require 180MB on your hard drive.

Install to:
C:\Program Files\Java\jdk1.8.0_271\

Step 6) Once you install Java in windows, click Close



Java SE Development Kit 8 Update 271 (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.

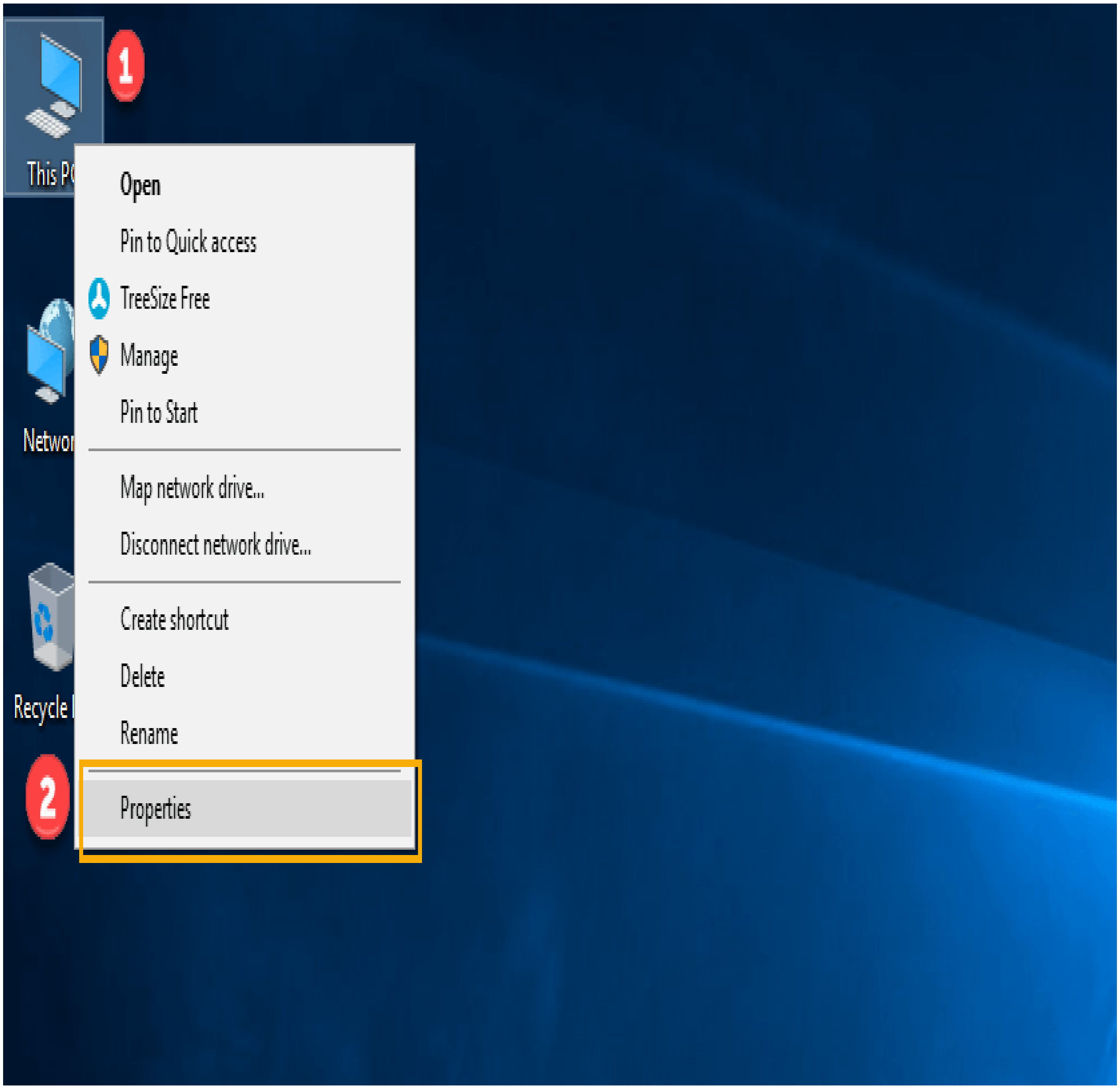
How to set Environment Variables in Java: Path and Classpath

The PATH variable gives the location of executables like javac, java etc. It is possible to run a program without specifying the PATH but you will need to give full path of executable like **C:\Program Files\Java\jdk1.8.0_271\bin\javac A.java** instead of simple **javac A.java**

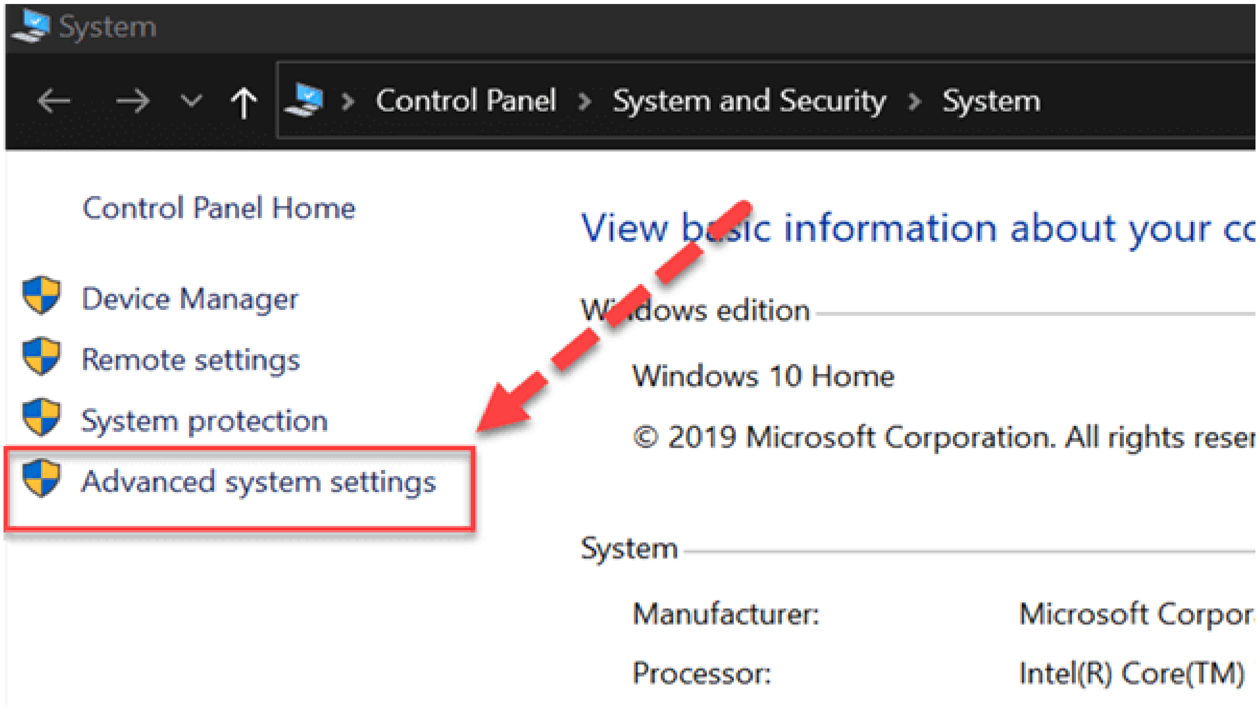
The CLASSPATH variable gives location of the Library Files.

Let's look into the steps to set the PATH and CLASSPATH

Step 1) Right Click on the My Computer and Select the properties



Step 2) Click on advanced system settings



View basic information about your cc

Windows edition

Windows 10 Home

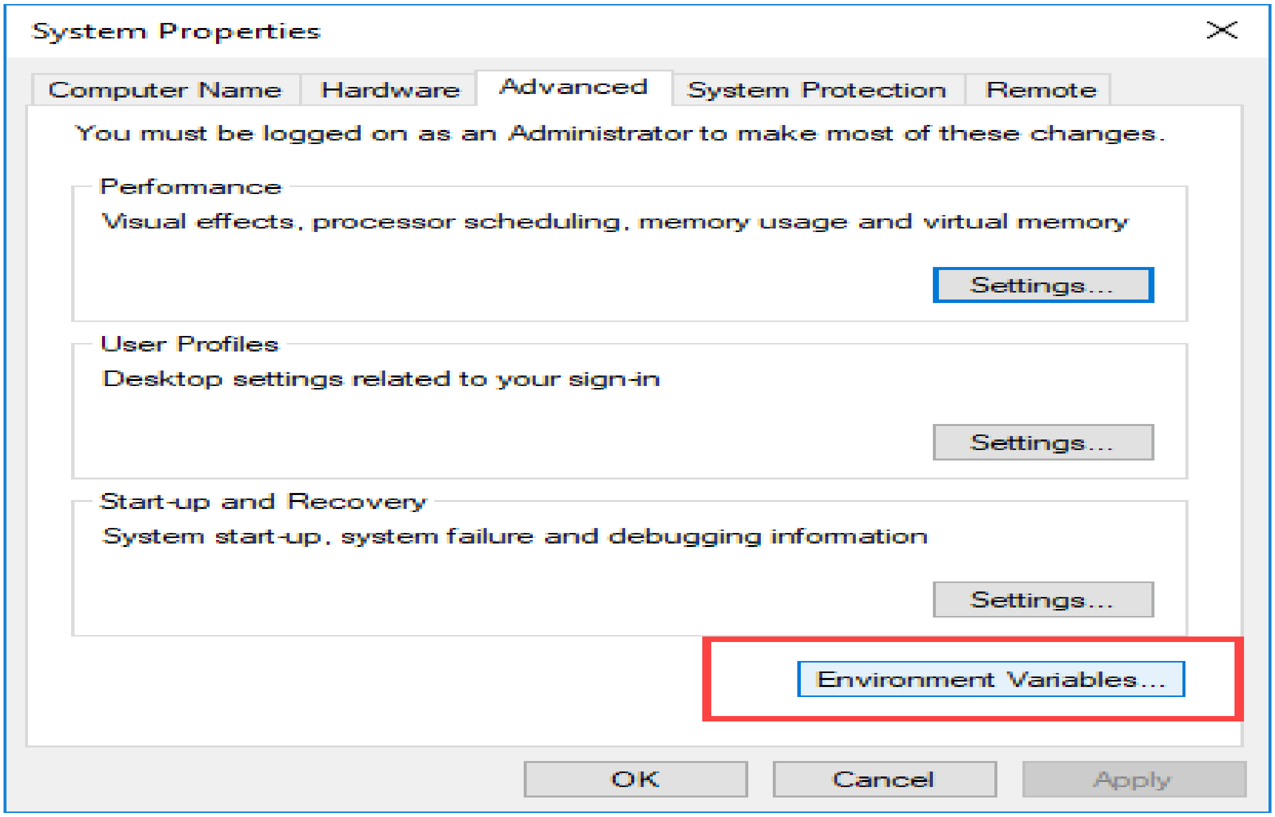
© 2019 Microsoft Corporation. All rights reser

System

Manufacturer: Microsoft Corpor

Processor: Intel(R) Core(TM)

Step 3) Click on Environment Variables to set Java runtime environment



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage and virtual memory

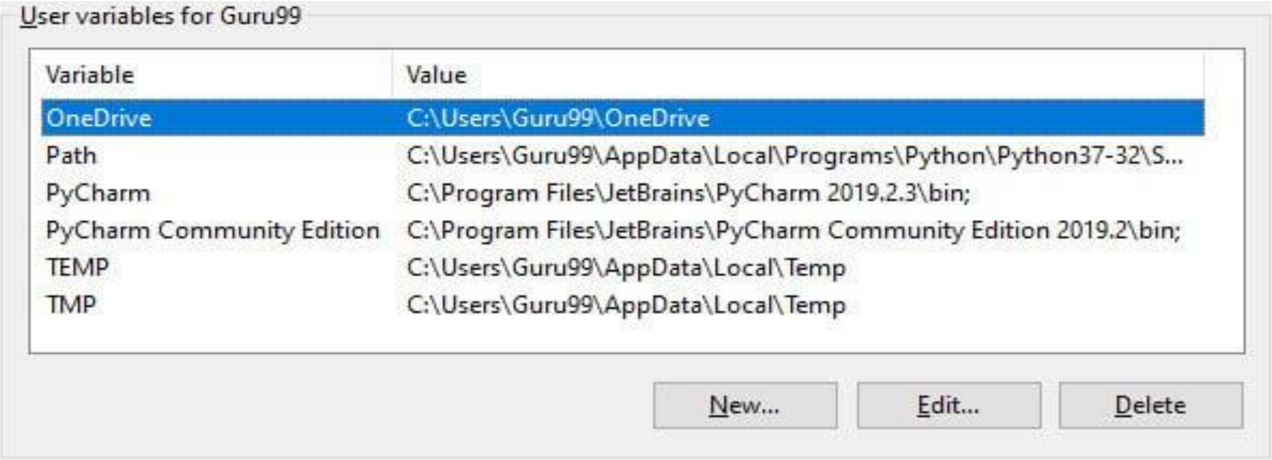
User Profiles

Desktop settings related to your sign-in

Start-up and Recovery

System start-up, system failure and debugging information

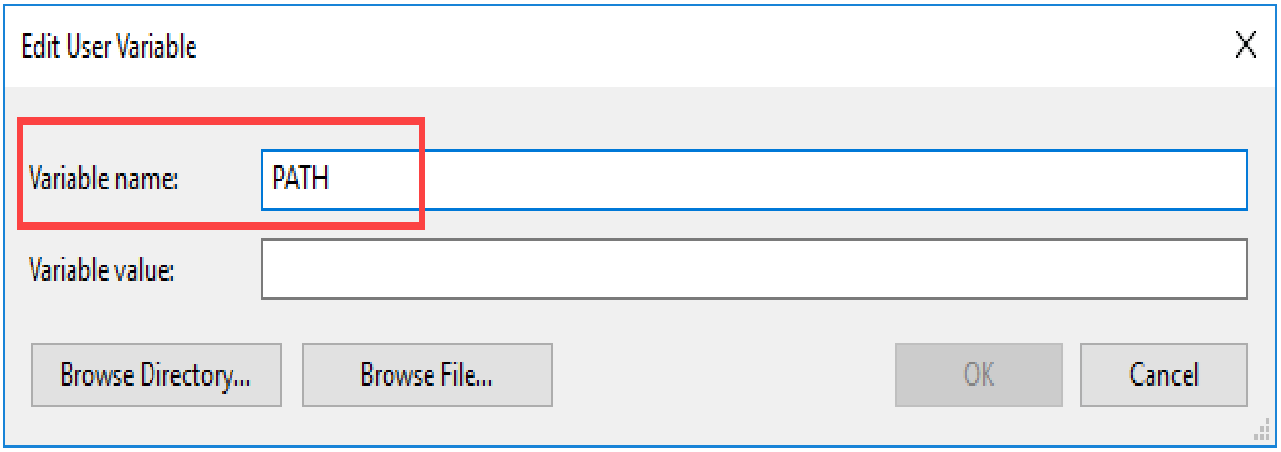
Step 4) Click on new Button of User variables



User variables for Guru99

Variable	Value
OneDrive	C:\Users\Guru99\OneDrive
Path	C:\Users\Guru99\AppData\Local\Programs\Python\Python37-32\S...
PyCharm	C:\Program Files\JetBrains\PyCharm 2019.2.3\bin;
PyCharm Community Edition	C:\Program Files\JetBrains\PyCharm Community Edition 2019.2\bin;
TEMP	C:\Users\Guru99\AppData\Local\Temp
TMP	C:\Users\Guru99\AppData\Local\Temp

Step 5) Type PATH in the Variable name.

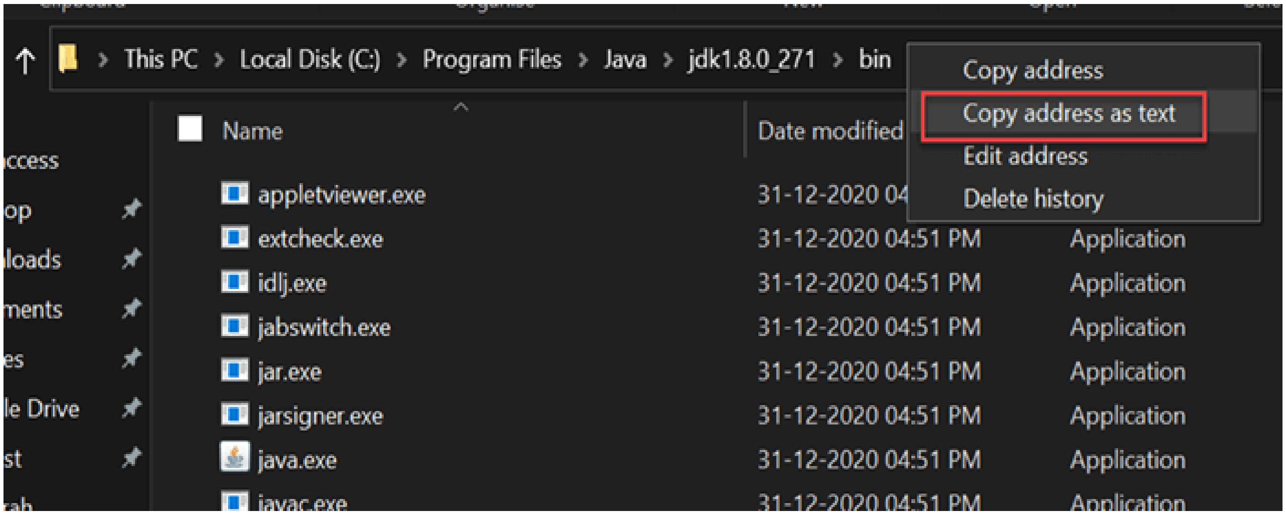


Edit User Variable

Variable name: PATH

Variable value:

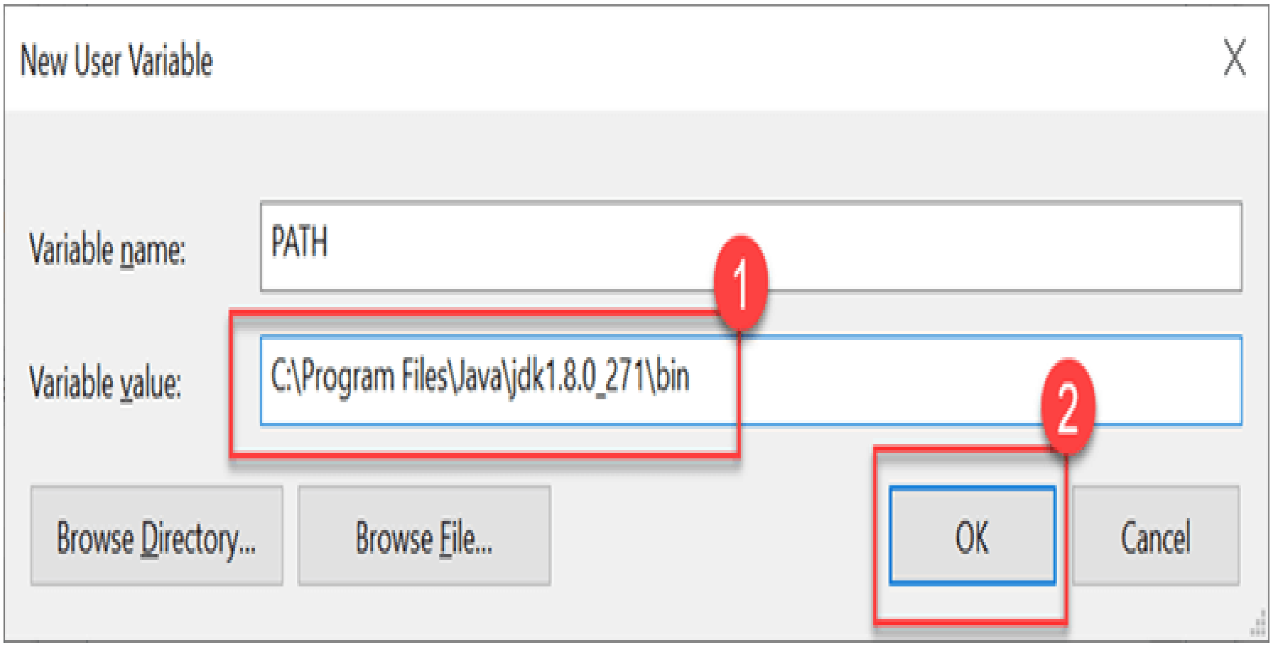
Step 6) Copy the path of bin folder which is installed in JDK folder.



This PC › Local Disk (C:) › Program Files › Java › jdk1.8.0_271 › bin

Name	Date modified	
appletviewer.exe	31-12-2020 04	
extcheck.exe	31-12-2020 04:51 PM	Application
idlj.exe	31-12-2020 04:51 PM	Application
jabswitch.exe	31-12-2020 04:51 PM	Application
jar.exe	31-12-2020 04:51 PM	Application
jarsigner.exe	31-12-2020 04:51 PM	Application
java.exe	31-12-2020 04:51 PM	Application
javac.exe	31-12-2020 04:51 PM	Application

Step 7) Paste Path of bin folder in Variable value. Click on OK Button.



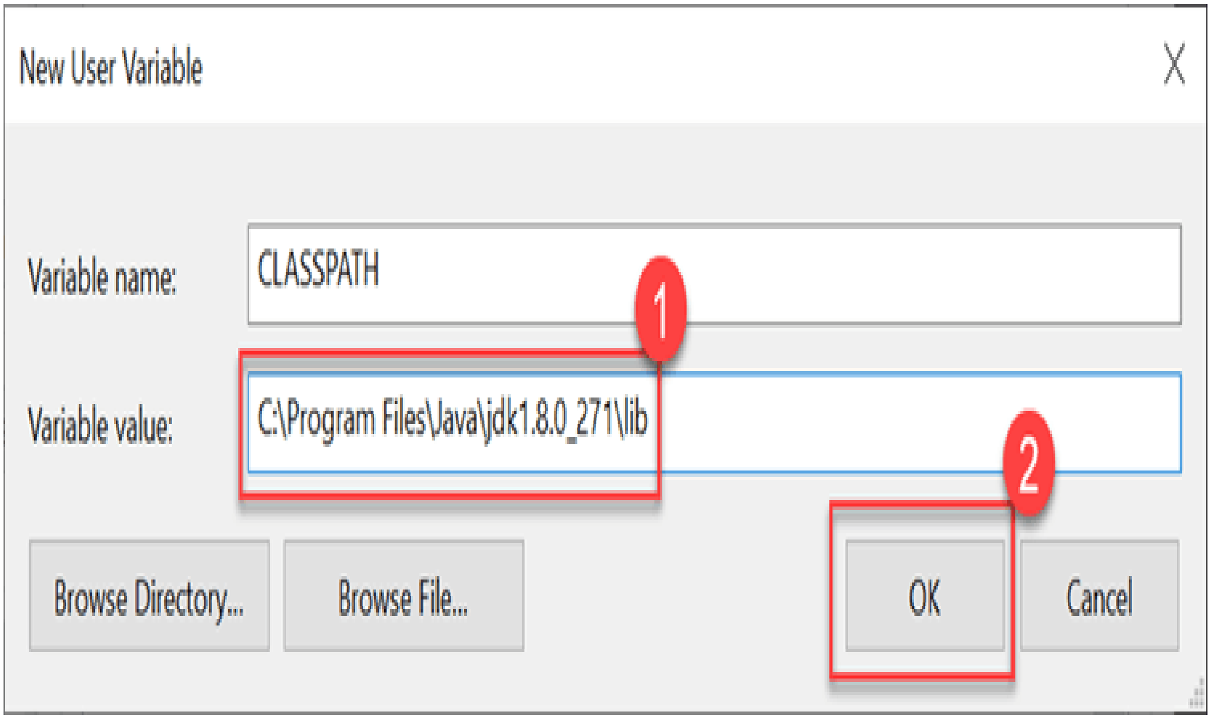
Variable name: PATH

Variable value: C:\Program Files\Java\jdk1.8.0_271\bin

Note: In case you already have a PATH variable created in your PC, edit the PATH variable to

PATH = <JDK installation directory>\bin;%PATH%;
Here, %PATH% appends the existing path variable to our new value

Step 8) You can follow a similar process to set CLASSPATH.



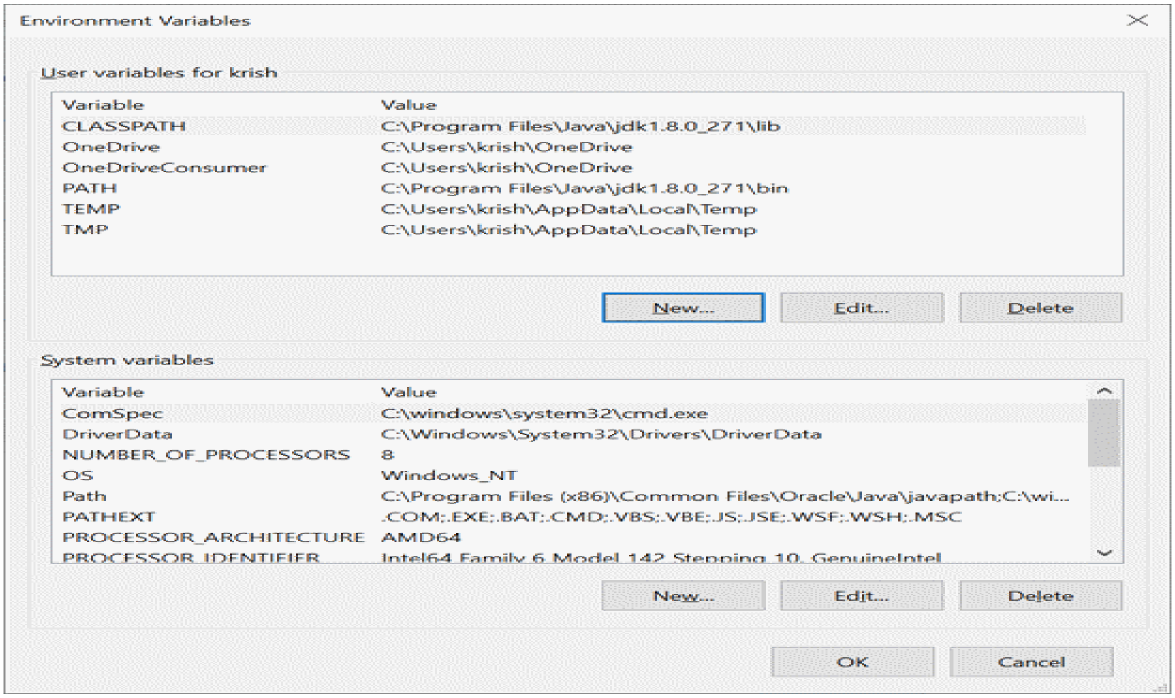
Variable name: CLASSPATH

Variable value: C:\Program Files\Java\jdk1.8.0_271\lib

Note: In case you java installation does not work after installation, change classpath to

CLASSPATH = <JDK installation directory>\lib\tools.jar;

Step 9) Click on OK button

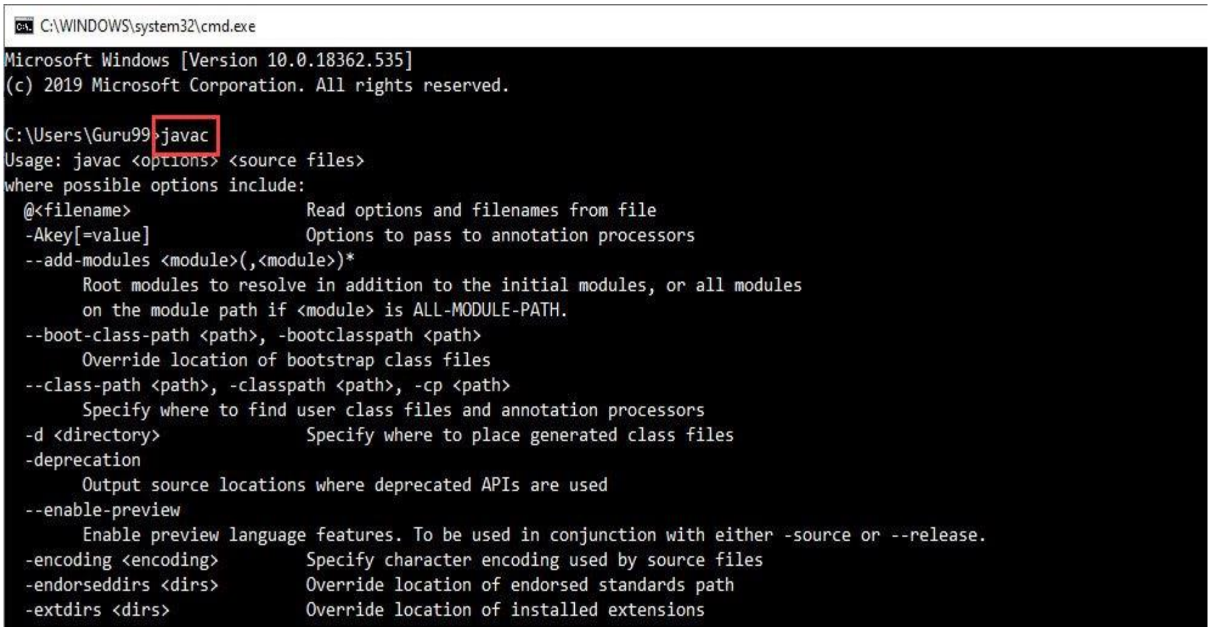


Variable	Value
CLASSPATH	C:\Program Files\Java\jdk1.8.0_271\lib
OneDrive	C:\Users\krish\OneDrive
OneDriveConsumer	C:\Users\krish\OneDrive
PATH	C:\Program Files\Java\jdk1.8.0_271\bin
TEMP	C:\Users\krish\AppData\Local\Temp
TMP	C:\Users\krish\AppData\Local\Temp

Variable	Value
ComSpec	C:\windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\wi...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 142 Stepping 10, GenuineIntel

Step 10) Go to command prompt and type javac commands.

If you see a screen like below, Java is installed.



```
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Guru99>javac
Usage: javac <options> <source files>
where possible options include:
  @<filename>                  Read options and filenames from file
  -Akey[=value]                Options to pass to annotation processors
  --add-modules <module>(,<module>)*
        Root modules to resolve in addition to the initial modules, or all modules
        on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
        Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
        Specify where to find user class files and annotation processors
  -d <directory>               Specify where to place generated class files
  -deprecation
        Output source locations where deprecated APIs are used
  --enable-preview
        Enable preview language features. To be used in conjunction with either -source or --release.
  -encoding <encoding>         Specify character encoding used by source files
  -endorseddirs <dirs>         Override location of endorsed standards path
  -extdirs <dirs>              Override location of installed extensions
```

1.(a) Write a java program to find the Fibonacci Series of a given number.

Program:

```
public class FactExample
{
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter the number: ");
int num=sc.nextInt();
int i=1,fact=1;
while(i<=num)
{
fact=fact*i;
i++;
}
System.out.println("Factorial of the number: "+fact);
}
}
```

Output:

```
Enter the number:
7
Factorial of the number: 5040
```

Result:

(b) Write a java program to perform Exchange of Two Numbers.

Program:

```
import java.util.*;
class SwapInteger
{
public static void main(String args[])
{
Scanner inp=new Scanner(System.in);
System.out.print("\n Enter First Number: ");
int a=inp.nextInt();
System.out.print("\n Enter Second Number: ");
int b=inp.nextInt();
System.out.println("Before Swapping: ");
System.out.println("First Number: " +a);
System.out.println("Second Number: " +b);
a=a+b;
b=a-b;
a=a-b;
System.out.println("After Swapping: ");
System.out.println("First Number: " +a);
System.out.println("Second Number: " +b);
}
}
```

Output:

```
Enter First Number: 4
Enter Second Number: 5
Before Swapping:
First Number: 4         Second Number: 5
After Swapping:
First Number: 5         Second Number: 4
```

Result:

Viva voce:

1. Who is the father of java? Why is Java a platform independent language.

2. Define public static void main(String args[]) in Java.

3. Why Java is not 100% Object-oriented.

4. Differentiate JDK, JRE and JVM.

5. What are the various access specifiers for Java classes.

2.(a) Write a java program to find the Greatest Among Two Numbers using Conditional Operator.

Program:

```
import java.util.*;
class ConditionalOperator
{
public static void main(String args[])
{
Scanner inp1=new Scanner(System.in);
System.out.print("\n Enter First Number: ");
int n1=inp1.nextInt();
Scanner inp2=new Scanner(System.in);
System.out.print("\n Enter Second Number: ");
int n2=inp2.nextInt();
int max;
max=(n1>n2)?n1:n2;
System.out.println("Greatest Among Two Numbers is :" +max);
}
}
```

Output:

```
Enter First Number: 5
 Enter Second Number: 10
Greatest Among Two Numbers is :10
```

Result:

(b) Write a java program to check whether the number is a palindrome or not.

Program:

```
import java.util.*;
class Palindrome
{
public static void main(String args[])
{
Scanner inp = new Scanner(System.in);
System.out.print("\n Enter Number: ");
int n = inp.nextInt();
int a,s = 0,m = n;
while (n != 0) // Extracting each digits and accumulating its sum.
{
a = n % 10;
s = s * 10 + a;
n = n / 10;
}
if (m == s) // Checking if reverse and original are identical.
System.out.println(m + " is a Palindrome Number");
else
System.out.println(m + " is not a Palindrome Number");
}
}
```

Output:

```
Enter Number: 141
141 is a Palindrome Number
```

Result:

Viva voce:

1. Pointers are used in C/ C++. Why does Java not make use of pointers.

2. What is JIT compiler in Java.

3. What is Object Oriented Programming.

4. What are Loops in Java? What are three types of loops.

5. What is an infinite Loop? How infinite loop is declared.

3.(a) Write a java program to find the reverse of a given number.

Program:

```
import java.util.*;
class NumReverse
{
public static void main(String args[])
{
Scanner inp=new Scanner(System.in);
System.out.print("\n Enter Number: ");
int n=inp.nextInt();
int a,s=0,m=n;
while(n!=0)   // Extracting each digits and accumulating its sum.
{
a=n%10;
s=s*10+a;
n=n/10;
}
System.out.println("Reverse of "+m+" is: "+s);
}
}
```

Output:

```
Enter Number: 432
Reverse of 432 is: 234
```

Result:

(b) Write a java program to check whether the number is a Prime or Not.

Program:

```
import java.util.*;
class PrimeNumber
{
public static void main(String args[])
{
Scanner inp=new Scanner(System.in);
System.out.print("\n Enter Number: ");
int n=inp.nextInt();
int i,c=0;
for(i=1;i<=n;i++)
{
if(n%i==0)
c++;
}
if(c==2)
System.out.println(n+" is a Prime Number");
else
System.out.println(n+" is not a Prime Number");
}
}
```

Output:

```
Enter Number: 5
5 is a Prime Number
```

Result:

<u>Viva voce:</u>

1. What is the difference between continue and break statement.

2. What’s the base class in Java from which all classes are derived.

3. Can main() method in Java can return any data.

4. what is Java Bytecode.

5. What is the difference between this() and super() in Java.

6**.**What is a classloader in Java.

4.(a) Write a java program to find the Factorial of a given Number using Recursion.

Program:

```
import java.util.*;
public class RecursionFact
{
static int factorial(int n)
{
if (n == 1)
return 1;
else
return(n * factorial(n-1));
}
public static void main(String args[])
{
Scanner inp1=new Scanner(System.in);
System.out.print("\n Enter n value: ");
int n=inp1.nextInt();
System.out.println("Factorial of a given number is: "+factorial(n));
}
}
```

Output:

```
Enter n value: 7
Factorial of a given number is: 5040
```

Result:

(b) Write a java program to find Fibonacci Series for a given number using Recursion.

Program:

```
import java.util.*;
public class RecursionFib
{
static int n1=0,n2=1,n3=0;
static void printFibo(int n)
{
if(count>0)
{
n3 = n1 + n2;
n1 = n2;
n2 = n3;
System.out.print(" "+n3);
printFibo(n-1);
}
}
public static void main(String args[])
{
Scanner inp1=new Scanner(System.in);
System.out.print("\n Enter n value: ");
int n=inp1.nextInt();
System.out.print(n1+" "+n2);
printFibo(count-2);
}
}
```

Output:

```
Enter n value: 7
0 1 1 2 3 5 8
```

Result

(c) Write a java program to implement Towers of Hanoi using Recursion.

Program:

```
import java.util.*;
class Towersofhanoi
{
static void towerOfHanoi(int n, char from_rod, char to_rod, char aux_rod)
{
if(n == 1)
{
System.out.println("Move disk 1 from rod " +  from_rod + " to rod " + to_rod);
return;
}
towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
System.out.println("Move disk " + n + " from rod " +  from_rod + " to rod " + to_rod);
towerOfHanoi(n-1, aux_rod, to_rod, from_rod);
}
public static void main(String args[])
{
Scanner inp1=new Scanner(System.in);
System.out.print("\n Enter n value: ");
int n=inp1.nextInt();
towerOfHanoi(n, 'A', 'C', 'B');
}
}
```

Output:

```
Enter n value: 4
Move disk 1 from rod A to rod B
Move disk 2 from rod A to rod C
Move disk 1 from rod B to rod C
Move disk 3 from rod A to rod B
Move disk 1 from rod C to rod A
Move disk 2 from rod C to rod B
Move disk 1 from rod A to rod B
Move disk 4 from rod A to rod C
Move disk 1 from rod B to rod C
Move disk 2 from rod B to rod A
Move disk 1 from rod C to rod A
Move disk 3 from rod B to rod C
Move disk 1 from rod A to rod B
Move disk 2 from rod A to rod C
Move disk 1 from rod B to rod C
```

Result:

Viva voce:

1.Define Recursion. List the advantages of Recursion.

2. What do you understand by an instance variable and a local variable.

3. Can you tell the difference between equals() method and equality operator (==) in Java.

4. What is the main objective of garbage collection.

5. Can we declare the main method of our class as private.

5.(a) Write a java program to find the Maximum and Minimum Number for a set of numbers.

Program:

```
import java.util.*;
class ArrMinMax
{
public static void main(String args[])
{
Scanner inp=new Scanner(System.in);
System.out.print("\n Enter Size of Array: ");
int n=inp.nextInt();
int i;
int arr[]=new int[n];

for(i=0;i<n;i++)
{
System.out.print("\n Enter The Element: ");
arr[i]=inp.nextInt();
}
```

Output:

```
Enter Size of Array: 6
Enter The Element: 2
 Enter The Element: 6
 Enter The Element: 1
 Enter The Element: 8
 Enter The Element: 10
 Enter The Element: 0
 Maximum Number: 10
 Minimum Number: 0
```

Result:

(b) Write a java program to perform Sum of Array Numbers.

Program:

```
import java.util.*;
class ArrSumOfNum
{
public static void main(String args[])
{
Scanner inp=new Scanner(System.in);
System.out.print("\n Enter Number of Numbers to be Calculated: ");
int n=inp.nextInt();
int i,sum=0;
int arr[]=new int[n];   //Creating N-size Array
for(i=0;i<n;i++)   //Entering N numbers in array
{
System.out.print("\n Enter: ");
arr[i]=inp.nextInt();
}
System.out.println();
for(i=0;i<n;i++)
{
System.out.print(arr[i]);
sum=sum+arr[i];       //Cumulative Sum
if(i<(n-1))
 System.out.print(" + ");
else
 System.out.print( " = ");
}
System.out.print(sum);
}
}
```

Output:

```
Enter Number of Numbers to be Calculated: 6
Enter: 1
 Enter: 4
 Enter: 6
 Enter: 8
 Enter: 9
 Enter: 23
1 + 4 + 6 + 8 + 9 + 23 = 51
```

Result:

Viva voce:

1. Define Array. List the different types of Arrays.

2. Can you pass the negative number in Array size.

3. Can you declare an Array without Array size.

4. Where does Array stored in JVM memory .

5. Which is legal int[] arr or int arr[].

6. How do you declare a two dimensional Array in java.

6.(a) Write a java to perform Substraction of Two matrices.

Program:

```
import java.util.*;
class SubtractMatrix
{
public static void main(String args[])
{
Scanner inp=new Scanner(System.in);
int r,c,i,j;
System.out.print("\n Enter Dimensions of Matrix (Row * Column) :\n ");
r=inp.nextInt();
c=inp.nextInt();
int a[][]=new int[r][c];     //Creating Matrices of size r*c
int b[][]=new int[r][c];
System.out.println("Enter into First Matrix: ");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
System.out.print("\n Enter: ");
a[i][j]=inp.nextInt();
}
}
System.out.println("Enter into Second Matrix: ");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
System.out.print("\n Enter: ");
b[i][j]=inp.nextInt();
}
}
int diff[][]=new int[r][c];
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
diff[i][j]=a[i][j]-b[i][j];     //Calculating difference of corresponding elements in both matrix
}
}
System.out.println();
System.out.println("First Matrix: ");    // Displaying Result
display(a,r,c);
System.out.println("Second Matrix: ");
display(b,r,c);   //Function to display matrix when invoked.
System.out.println("Resultant Matrix after Subtraction: ");
display(diff,r,c);
}
public static void display(int arr[][],int row,int col)
{
int i,j;
for(i=0;i<row;i++)
{
for(j=0;j<col;j++)
{
System.out.print(arr[i][j]+" ");
}
System.out.println();
}
```

```
}
}
```

Output:

```
Enter Dimensions of Matrix (Row * Column) :
 3
 3
Enter into First Matrix:
 Enter: 1
 Enter: 2
 Enter: 3
 Enter: 4
 Enter: 5
 Enter: 6
 Enter: 7
 Enter: 8
 Enter: 9
Enter into Second Matrix:
 Enter: 6
 Enter: 7
 Enter: 5
 Enter: 4
 Enter: 3
 Enter: 2
 Enter: 7
 Enter: 8
 Enter: 9
First Matrix:
1 2 3
4 5 6
7 8 9
```

```
Second Matrix:
6 7 5
4 3 2
7 8 9
Resultant Matrix after Subtraction:
-5 -5 -2
0   2   4
0   0   0
```

<u>Result:</u>

(b) Write a java program to find the Transpose of a Matrix.

Program:

```
import java.util.*;
class TransposeMatrix
{
public static void main(String args[])
{
Scanner inp=new Scanner(System.in);
int r,c,i,j;
System.out.print("\n Enter Dimensions of Matrix (Row * Column) :\n ");
r=inp.nextInt();
c=inp.nextInt();
int a[][]=new int[r][c];     //Creating Matrices of size r*c
System.out.println("Enter into First Matrix: ");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
System.out.print("\n Enter: ");
a[i][j]=inp.nextInt();
}
}
int trans[][]=new int[r][c];
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
trans[i][j]=a[j][i];     //Calculating Transpose of a matrix
}
}
System.out.println();
System.out.println("First Matrix: ");    // Displaying Result
display(a,r,c);
System.out.println("Resultant Matrix after Transpose: ");
display(trans,r,c);
}
public static void display(int arr[][],int row,int col)
{
int i,j;
for(i=0;i<row;i++)
{
for(j=0;j<col;j++)
{
System.out.print(arr[i][j]+" ");
}
System.out.println();
}
}
}
```

Output:

```
Enter Dimensions of Matrix (Row * Column) :
 3
 3
Enter into First Matrix:
Enter: 1
Enter: 2
Enter: 3
Enter: 4
Enter: 5
```

```
Enter: 6
Enter: 78
Enter: 9
Enter: 3
First Matrix:
1 2 3
4 5 6
78 9 3
Resultant Matrix after Transpose:
1 4 78
2 5 9
3 6 3
```

Result:

Viva voce:

1. What are the advantages of Array.

2. What are the disadvantages of Array.

3. What are jagged arrays in java.

7. Write a java program to perfrom Multiplication of Two matrices.

Program:

```
import java.util.*;
class ProductMatrix
{
public static void main(String args[])
{
Scanner inp=new Scanner(System.in);
int r1,c1,r2,c2,i,j,k;
System.out.print("\n Enter Dimensions of First Matrix (Row * Column) : \n");
r1=inp.nextInt();
c1=inp.nextInt();
System.out.print("\n Enter Dimensions of Second Matrix (Row * Column) : \n");
r2=inp.nextInt();
c2=inp.nextInt();
if(c1!=r2)
{
 System.out.println("Number of Columns of First Matrix should be equal to Number of Rows of Second Matrix");
 System.exit(0);
}
int a[][]=new int[r1][c1];
System.out.println("Enter into First Matrix: ");
for(i=0;i<r1;i++)
{
for(j=0;j<c1;j++)
{
System.out.print("\n Enter: ");
a[i][j]=inp.nextInt();
}
```

```
}
int b[][]=new int[r2][c2];
System.out.println("Enter into Second Matrix: ");
for(i=0;i<r2;i++)
{
for(j=0;j<c2;j++)
{
System.out.print("\n Enter: ");
b[i][j]=inp.nextInt();
}
}
int pro[][]=new int[r1][c2];
for(i=0;i<r1;i++)
{
for(j=0;j<c2;j++)
{
for(k=0;k<c1;k++)
{
pro[i][j]=pro[i][j]+(a[i][k]*b[k][j]);
}
}
}
System.out.println("\n First Matrix: ");
for(i=0;i<r1;i++)
{
for(j=0;j<c1;j++)
{
System.out.print(a[i][j]+" ");
}
System.out.println();
```

```
}
System.out.println("\n Second Matrix: ");
for(i=0;i<r2;i++)
{
for(j=0;j<c2;j++)
{
System.out.print(b[i][j]+" ");
}
System.out.println();
}
System.out.println("\n Product Matrix: ");
for(i=0;i<r1;i++)
{
for(j=0;j<c2;j++)
{
System.out.print(pro[i][j]+" ");
}
System.out.println();
}
}
}
```

Output:

```
Enter Dimensions of First Matrix (Row * Column) :
3
3
Enter Dimensions of Second Matrix (Row * Column) :
3
3
Enter into First Matrix:
Enter: 1
Enter: 2
Enter: 3
Enter: 4
Enter: 5
Enter: 6
Enter: 7
Enter: 8
Enter: 9
Enter into Second Matrix:
Enter: 1
Enter: 0
Enter: 0
Enter: 0
Enter: 1
Enter: 0
Enter: 0
Enter: 0
Enter: 1
First Matrix:
1 2 3
```

```
4 5 6
7 8 9
Second Matrix:
1 0 0
0 1 0
0 0 1
Product Matrix:
1 2 3
4 5 6
7 8 9
```

Result:

Viva voce:

1. How to convert ArrayList to String Array in java.

2. What is the default value of Array for different data types.

3. Can you change size of Array in java after creation.

8.(a) Write a java program to find the Reverse of a given String.

Program:

```
import java.util.*;
class ReverseString
{
public static void main(String args[])
{
Scanner inp=new Scanner(System.in);
System.out.print("\n Enter String: ");
String s=inp.nextLine();
int k=s.length();
int i;
char c;
String z="";
for(i=(k-1);i>=0;i--)
{
c=s.charAt(i);
z=z+c;
}
System.out.println("Reversed String: "+z);
}
}
```

Output:

```
Enter String: Prasanth
Reversed String: htnasarP
```

Result:

(b) Write a java program to check whether the strins are Anagrams or Not.

Program:

```
import java.util.*;
class Anagrams
{
public static void main(String args[])
{
Scanner inp=new Scanner(System.in);
System.out.print("\n Enter First String: ");
String s1=inp.nextLine();
System.out.print("\n Enter Second String: ");
String s2=inp.nextLine();
int n1=0,n2=0,i;
for(i=0;i<s1.length();i++)
 n1=n1+s1.charAt(i);
 for(i=0;i<s2.length();i++)
 n2=n2+s2.charAt(i);
 if(n1==n2)
  System.out.println(s1+" and "+s2+" are anagrams.");
else
  System.out.println(s1+" and "+s2+" are not anagrams.");
}
}
```

Output:

```
Enter First String: prasanth
Enter Second String: prakash
prasanth and prakash are not anagrams.
C:\Users\prasa>java Anagrams
Enter First String: prasanth
Enter Second String: prasanth
prasanth and prasanth are anagrams.
```

Result:

Viva voce:

1. What are the different ways to create string objects.

2. Difference between String, StringBuffer, and StringBuilder in java.

3. How to compare two strings in java.

4. Difference between string in C and string in java.

9.(a) Write a java program to create student class and display student details.

Program:

```
public class Student
{
String name;
int rollno;
int age;
void info()
{
System.out.println("Name: "+name);
System.out.println("Roll Number: "+rollno);
System.out.println("Age: "+age);
}
public static void main(String args[])
{
Student student = new Student();
// Accessing and property value
student.name = "Prasanth";
student.rollno = 557;
student.age = 25;
// Calling method
student.info();
}
}
```

Output:

```
Name: Prasanth
Roll Number: 557
Age: 25
```

Result:

(b) Write a java program to Concatenate two Strings.

Program:

```
import java.util.*;
class FuncConcat
{
public static void main(String args[])
{
Scanner inp=new Scanner(System.in);
System.out.print("\n Enter First String: ");
String s1=inp.nextLine();
System.out.print("\n Enter Second String: ");
String s2=inp.nextLine();
String z=s1.concat(s2);
System.out.println("Concatenated String: "+z);
}
}
```

Output:

```
Enter First String: prasanth
Enter Second String: sunny
Concatenated String: prasanthsunny
```

Result:

Viva voce:

1. List the Object-Oriented Concepts in java.

2.Define Class. How to create a Class in java.

3.Define object. How to Create an object in java.

4.Define Abstraction. write the Example.

5.Define Encapsulation. write the example.

6.Define Abstract Class in java.

10.(a) Write a java program to implement Single Inheritance.

Program:

```
class P
{
int a=10;
void display()
{
System.out.println(" Parent Class ");
}
}
class C extends P
{
int b=20;
void show()
{
System.out.println(" Child Class ");
}
}
class SingleInherit
{
public static void main(String args[])
{
C obj=new C();
System.out.println(obj.b);
obj.show();
System.out.println(obj.a);
obj.display();
}
}
```

Output:

```
20
Child Class
10
Parent Class
```

Result:

(b) Write a java program to implement Multilevel Inheritance.

Program:

```
class Gp
{
int a=10;
void display()
{
System.out.println("Grand parent class");
}
}
class P extends Gp
{
int b=20;
void print()
{
System.out.println("Parent Class");
}
}
class C extends P
{
int c=30;
void show()
{
System.out.println("Child class");
}
}
class MultipleInherit
{
public static void main(String args[])
{
C obj=new C();
System.out.println(obj.c);
obj.show();
System.out.println(obj.b);
obj.print();
System.out.println(obj.a);
obj.display();
}
}
```

Output:

```
30
Child class
20
Parent Class
10
Grandparent class
```

Result:

<u>Viva voce:</u>

1.Define Inheritance. Write the example.

2.Define Single Inheritance.

3. Define Multilevel Inheritance.

4.Differentiate Single and Multilevel Inheritance.

5.Which key word is used to implement Inheritance in java.

11.(a) Write a java program to implement Hierarchical Inheritance.

Program:

```
class P
{
int a=10;
void display()
{
System.out.println(" Parent class ");
}
}
class C extends P
{
int b=20;
void print()
{
System.out.println(" Child class 1 ");
}
}
class D extends P
{
int c=30;
void show()
{
System.out.println(" Child class 2 ");
}
}
class Hierarchy
{
public static void main(String args[])
{
C obj1=new C();
System.out.println(obj1.b);
obj1.print();
D obj2=new D();
System.out.println(obj2.c);
obj2.show();
System.out.println(obj2.a);
obj2.display();
}
}
```

Output:

```
20
Child class 1
30
Child class 2
10
Parent class
```

Result:

(b) Write a java program to implement Multiple Inheritance using Interfaces Concept.

Program:

```
interface A
{
void show();
}
interface B
{
void display();
}
class AB implements A,B
{
public void show()
{
System.out.println(" A Show ");
}
public void display()
{
System.out.println(" B display ");
}
}
class Interface
{
public static void main(String args[])
{
AB obj = new AB() ;
obj.show();
obj.display();
}
}
```

Output:

```
A Show
B display
```

Result:

Viva voce:

1.Define Multiple and Hierarchical Inheritance.

2.Define Interface.

3.Howto implement Multiple and Hierarchical Inheritance in java.

4.List the keywords that are used to implement interfaces.

12.(a) Write a java program to implement Constructor Overloading.

Program:

```
class A
{
A()
{
System.out.println("NBKRIST");
}
A(int a)
{
System.out.println("a = " +a);
}
}
class COverload
{
public static void main(String args[])
{
A Obj1=new A() ;
A Obj2=new A(20) ;
}
}
```

Output:

```
NBKRIST
a = 20
```

Result:

(b) Write a java program to implement Method Overloading.

Program:

```
class A
{
void display()
{
System.out.println("NBKRIST");
}
void display(int a)
{
System.out.println("a = " +a);
}
}
class Methodoverload
{
public static void main(String args[])
{
A object = new A();
object.display();
object.display(10);
}
}
```

Output:

```
NBKRIST
a = 10
```

Result:

Viva voce:

1.Define Polymorphism. Write the Example.

2.List the different types of Polymorphism.

3.Define Overloading.

4.Define Constructor.

5.How to implement Compile time Polymorphism.

13.(a) Write a java program to implement Operator Overloading for Concatenation.

Program:

```
class C
{
int a=3,b=5;
public static void main(String args[])
{
C obj=new C();
System.out.println("a+b= " +obj.a +obj.b);
}
}
```

Output:

```
a+b= 35
```

Result:

(b) Write a java program to implement Operator Overloading for Addition operation.

Program:

```
class D
{
int a=3,b=5;
public static void main(String args[])
{
D obj=new D();
System.out.println("a+b= " +(obj.a +obj.b));
}
}
```

Output:

```
a+b= 8
```

Result:

<u>Viva voce:</u>

1.Differentiate Constructor and Method.

14.Write a java program to implement Method Overriding.

Program:

```
class Parent
{
 void display()
{
System.out.println(" NBKRIST ");
}
}
class Child  extends Parent
{
void display()
{
System.out.println(" CSE ");
}
}
class Override
{
public static void main(String args[])
{
Child a = new Child();
a.display();
}
}
```

Output:

```
CSE
```

Result:

Viva voce:

1.How to implement Run time Polymorphism.

2.Define Overriding.

3.Which concept is used to implement Overriding.

4.In which cases Overriding is not possible.

15. Write a java program to implement Exception Handling Mechanism.

Program 1:

```
class A
{
void display()
{
try
{
int a = 5/0;
}
catch(ArithmeticException e)
{
System.out.println(e);
}
System.out.println(" Exception Handled");
}
}
class Except
{
public static void main(String args[])
{
A obj=new A();
obj.display();
}
}
```

Output:

```
java.lang.ArithmeticException: / by zero
Exception Handled
```

Result:

Program 2:

```
class A
{
void display()
{
try
{
int a[] = new int[5];
a[6]=5;
}
catch(ArrayIndexOutOfBoundsException e)
{
System.out.println(e);
}
System.out.println(" Exception Handled ");
}
}
class Except1
{
public static void main(String args[])
{
A obj=new A();
obj.display();
}
}
```

Output:

```
java.lang.ArrayIndexOutOfBoundsException: Index 6 out of bounds for length 5
 Exception Handled
```

Result:

Program 3:

```
class A
{
void display()
{
try
{
String str ="Hello";
int num = Integer.parseInt(str) ;
}
catch (NumberFormatException e)
{
System. out. println(e);
}
System.out.println(" Exception Handled ");
}
}
class Except2
{
public static void main(String args[])
{
A obj  = new  A();
obj.display();
}
}
```

Output:

```
java.lang.NumberFormatException: For input string: "Hello"
Exception Handled
```

Result:

Program 4:

```
class A
{
void display()
{
try
{
String str=null;
System.out.println(str.length());
}
catch(NullPointerException e)
{
System.out.println(e);
}
System.out.println("Exception Handled");
}
}
class Except3
{
public static void main(String args[])
{
A obj=new A();
obj.display();
}
}
```

Output:

```
java.lang.NullPointerException: Cannot invoke "String.length()" because "<local1>" is null
Exception Handled
```

Result:

Program 5:

```
class Demo
{
public static void main(String args[])
{
int a[]=new int[2];
try
{
System.out.println("a[4]= "+a[ 4]);
}
catch ( Exception e)
{
System.out.println(e);
}
finally
{
a[1]=30 ;
System.out.println("a[1]= " +a[1]);
}
System.out.println("End of the program");
}
}
```

Output:

```
java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 2
a[1]= 30
End of the program
```

Result:

Viva voce:

1.Define Exception.

2.What are the different types of Exceptions in java.

3.List the Compile time Exceptions.

4.List the Run time Exceptions.

5.List the Keywords to implement Exception Handling in Java.

6.What is the main purpose of try and catch blocks in java.

7.what is the main purpose of finally block in java.

16.(a) Write a java program to implement Implicit throw in Exception Handling.

Program:

```
class Demo1
{
public static void main(String args[])
{
int x,y;
int a=10,b=5,c=5;
try
{
x=a/(b-c) ;
System.out.println(x);
}
catch(Exception e)
{
System.out.println(e);
}
y=a/(b+c);
System.out.println(y);
}
}
```

Output:

```
java.lang.ArithmeticException: / by zero
1
```

Result:

(b) Write a java program to implement Explicit throw in Exception Handling.

Program:

```
class Demo2
{
public static void main(String args[])
{
try
{
throw new ArithmeticException("THROW DEMO") ;
}
catch(ArithmeticException e)
{
System.out.println(e);
}
}
}
```

Output:

```
java.lang.ArithmeticException: THROW DEMO
```

Result:

(c) Write a java program to implement throws Keyword in Exception Handling.

Program:

```
public class Main
{
static void checkAge(int age)throws ArithmeticException
{
if(age<18)
{
throw new ArithmeticException("Accessdenined - you must be atleast 18 years old");
}
else
{
System.out.println("Access grantees your age is enough");
}
}
public static void main(String args[])
{
checkAge(54);
}
}
```

Output:

```
Access grantees your age is enough
```

Result:

Viva voce:

1.What is the main purpose of throw keyword in java.

2.Differentiate Implicit and Explicit throw.

3.What is the main purpose of throws keyword in java.

17. Write a simple java program using Packages.

Program for Package Creation:

```
package pack;
public class PackDemo
{
public void show()
{
System.out.println("Welcome To NBKRIST");
}
}
```

Output:

```
C:\Users\prasa>javac -d . PackDemo.java
C:\Users\prasa>javac pack1.java
```

Program:

```
import pack.PackDemo;
class Pack1
{
public static void main(String args[])
{
PackDemo obj=new PackDemo();
obj.show();
}
}
```

Output:

```
C:\Users\prasa>javac Pack1.java
C:\Users\prasa>java Pack1
Welcome To NBKRIST
```

Result:

Viva voce:

1.Define package? Which keyword is used to create package.

2.Which package is imported, by default.

3.Can a class declare as private be accessed outside its package.

4.What are the types of packages in Java.

5.What is a standard package in Java.

6.What is a user-defined package in Java.

7.How to create a user-defined package.

8.How to compile a source code of Java that is created as a package.

9.What happens at background when “Javac –d. Employe.Java “is executed.

18.(a) Write a Simple java program to display simple message using Applets.

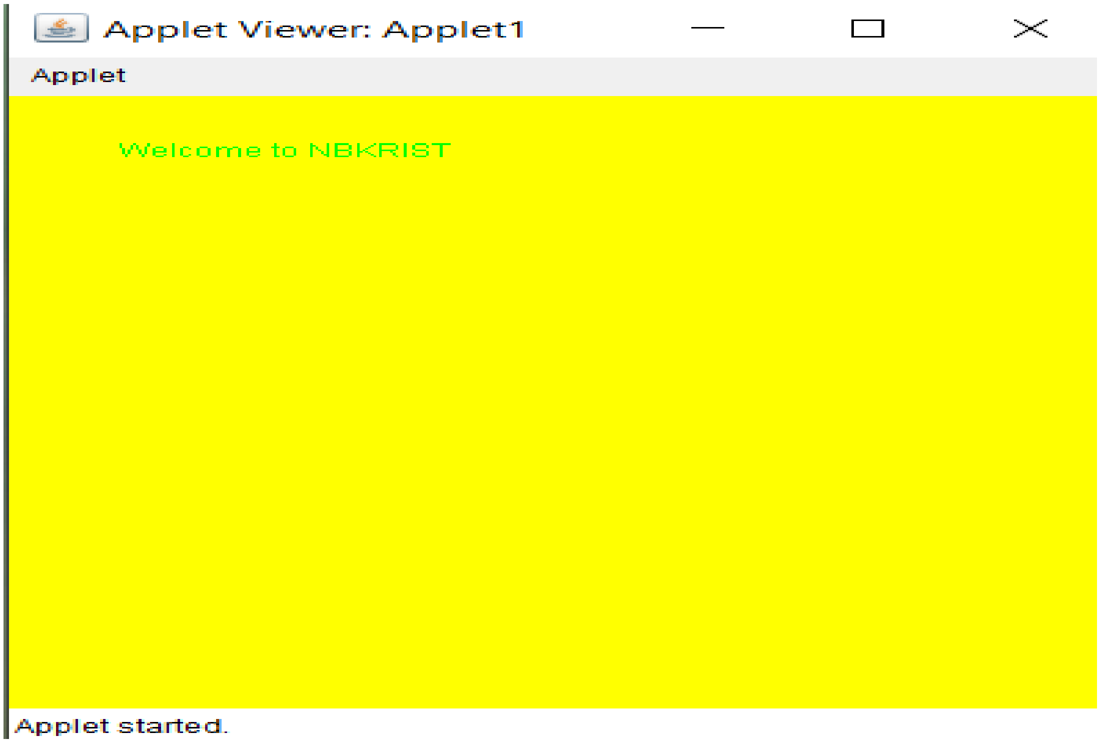
Program:

```
import java.applet.*;
import java.awt.*;
/*<applet code=Applet1 width=400 height=400>
</applet>*/
public class Applet1 extends Applet
{
public void init()
{
setBackground(Color.yellow);
setForeground(Color.green);
}
public void paint(Graphics g)
{
g.drawString("Welcome to NBKRIST",40,40);
}
}
```

Output:

```
C:\Users\prasa>javac Applet1.java

C:\Users\prasa>appletviewer Applet1.java
```



Welcome to NBKRIST

Applet started.

Result:

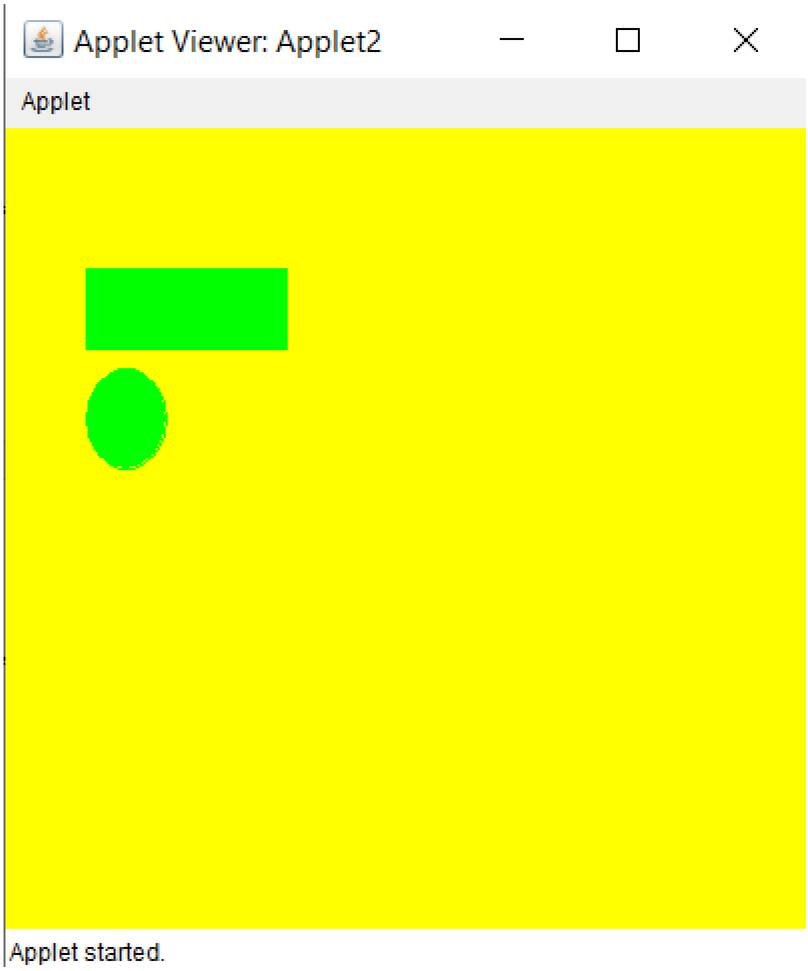
(b) Write a java program to implement Graphics class and methods in Applets.

Program:

```
import java.applet.*;
import java.awt.*;
/*<applet code=Applet2 width=400 height=400>
</applet>*/
public class Applet2 extends Applet
{
public void init()
{
setBackground(Color.yellow);
setForeground(Color.green);
}
public void paint(Graphics g)
{
g.drawRect(40,70,100,40);
g.fillRect(40,70,100,40);
g.drawOval(40,120,40,50);
g.fillOval(40,120,40,50);
}
}
```

Output:

```
C:\Users\prasa>javac Applet1.java
C:\Users\prasa>appletviewer Applet1.java
```



Result:

Viva voce:

1. What is an applet? How does applet differ from applications?

2. What are Methods that controls an applet’s life cycle.

3. What are the attributes of Applet tags.

4. How to set the background color within the applet area.