**Program No.1: Install and Setup java environment .Install java editor (Eclipse for Enterprise Java) and configure workspace. Execution of first java program. Java code execution process.**

Java environment setup:

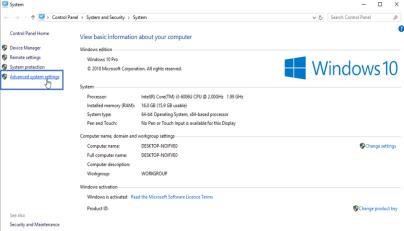
In today’s world, Java is one of the most popular programming language which is being used by most skilled professionals. However, using it on the command line is not feasible sometimes. Therefore, to overcome this, we can use it on Eclipse IDE. Let’s see how to setup Java environment on Eclipse IDE.

Step 1: Install Java

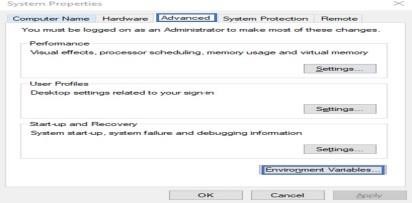
Step 2: Setup Eclipse IDE on Windows

Step 3: Hello World Program Install Java:

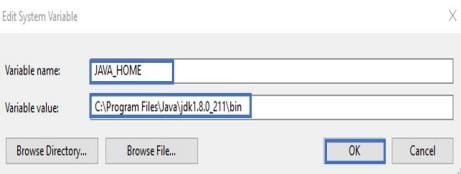
* Follow the below steps to complete your Java installation.
* Go to the Java Downloads Page and click on the option of Download.
* Now, once the file is downloaded, run the installer and keep clicking on Next, till you finally get a dialog box, which say, you have finished installing.
* Once the installation is over follow the below instructions to set the path of the file. Now right click on ThisPC/ My Computer Icon-> Go to its properties and its Advanced System Settings. Refer below.



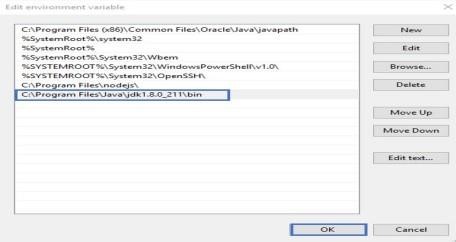
* Now, click on ‘Environment Variables’ under ‘Advanced’ tab as shown below:



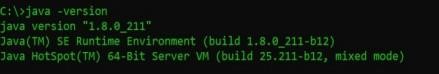
Next, under System Variables choose New. Enter the variable name as ‘JAVA\_HOME’ and the full path to Java installation directory as per your system as shown below:

 Next thing that you have to do is to configure your environment variables. Let’s see how to do that.

Here, you have to edit the path of the system variable as shown below. Then click OK.

Govt.

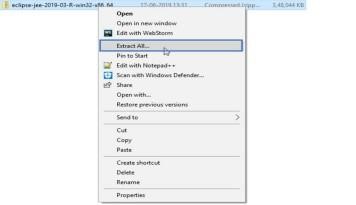
* Now to cross-check the installation, just run following command in cmd – java -version. It should display the installed version of Java in your system.



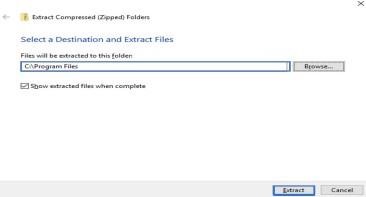
Install Eclipse:

Follow the below steps to configure Eclipse on your system:

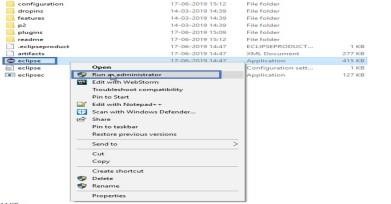
Step 1: Navigate to the following URL – https://www.eclipse.org/downloads/packages/ and select the download link depending on your system architecture – (Windows, Mac OS or Linux) and download it. Step 2: Once the download is over, extract the zipped file by right-clicking on the folder and choose Extract All. Refer below.



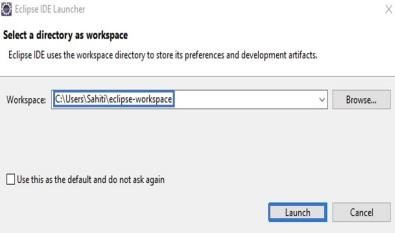
Step 3: You will be then redirected to a dialog box, where you have to choose the directory in which you wish to extract the files. Then click on Extract. Refer below.



Step 4: After extracting files, open the folder and launch eclipse.exe.

 Step 5:

Then, you have to choose the Launch directory for Eclipse and then click on Launch. Refer below.



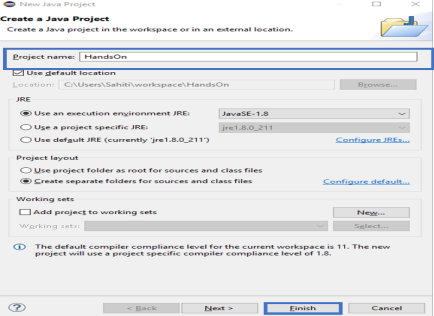
Step 6: Once Eclipse launches, you will see the below window:

Govt.

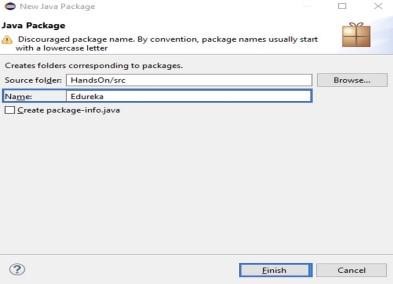
**Executing first Java Program : Hello World Program:**

Step 1: Launch Eclipse IDE and go to File ->New -> Java Project Step 2:

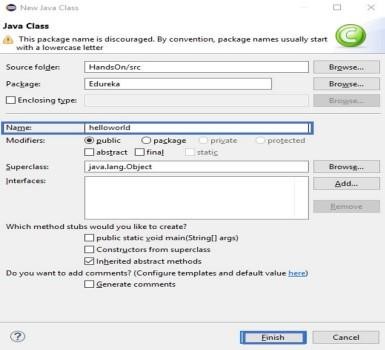
Mention the project name and click on Finish.

Step 3: Now,

go to the Project, Right-Click on the Project and choose Package. In the dialog box, which opens up, mention the Package name as below and click on Finish.



Step 4: Now, right click on the Package, go to New and choose Class. Mention the class name and click on Finish. Refer below.



Step 5: Now, mention the following code in the workspace. public class helloworld

|  |  |
| --- | --- |
|  | class HelloWorld  {  public static void main(String args[])  {  System.out.println("Hello world!");  }  } |

Step 6: Now, execute your file, by right-clicking on the helloworld.java file and choose Run As -> Java Application .

# Program No.2: Code, execute and debug programs that uses different types of

**Variables and datatypes;identify and resolve issues in the given code**

class Datatypes

{ public static void main(String[] args)

{ byte myByte1,myByte2;

myByte1 = 120; myByte2 = -48; System.out.println("Byte1: " +myByte1);

System.out.println("Byte2: " +myByte2); myByte1++; // Looping back within the range

System.out.println("Incremented Value of Byte1:" +myByte1);

short myShort = 6000;

System.out.println("\nShort:" +myShort);

int myInteger1, myInteger2, result;

myInteger1 = -7000; myInteger2 = 9000; result = myInteger1 + myInteger2;

System.out.println("\nInteger1:"+myInteger1);

System.out.println("Integer2:"+myInteger2);

System.out.println("Integer1 + Integer2: " +result);

long myLong1, myLong2, result1;

myLong1 = 100000000L; myLong2 = 200L;

result1 = myLong1 \* myLong2;

System.out.println("\nLong1: " +myLong1);

System.out.println("Long2: " +myLong2);

System.out.println("Long1 \* Long2: " +result1);

float myFloat1,myFloat2,result2;

myFloat1=1000.666f; myFloat2=110.77f; result2=myFloat1-myFloat2;

System.out.println("\nFloat1: "+myFloat1);

System.out.println("Float2: "+myFloat2);

System.out.println("Float1-Float2: "+result2);

double myDouble1, myDouble2, result3;

myDouble1 = 48976.8987; myDouble2 = 29513.7812d; result3 = myDouble1 + myDouble2;

System.out.println("\nDouble1: " +myDouble1);

System.out.println("Double2: " +myDouble2);

System.out.println("Double1 + Double2: " +result3); boolean myBool = true; if(myBool == true)

System.out.println("\nI am using a Boolean data type");

System.out.println(myBool);

char myChar1 = 'A'; char myChar2 = 66;

System.out.println("\nmyChar1: " +myChar1);

System.out.println("myChar2: " +myChar2); myChar2++; // valid increment operation

System.out.println("The Incremented value of myChar2: " +myChar2);

String string1 = "\nGPT ARAKERE"; // declaring string using string literal

System.out.println(string1);

}

}

**Output**:

Byte1: 120

Byte2: -48

Incremented Value of Byte1:121

Short: 6000

Integer1:-7000

Integer2:9000

Integer1 + Integer2: 2000

Long1: 100000000

Long2: 200

Long1 \* Long2: 20000000000

Float1: 1000.666

Float2: 110.77

Float1-Float2: 889.896

Double1: 48976.8987

Double2: 29513.7812

Double1 + Double2: 78490.6799 I am using a Boolean data type true

myChar1: A myChar2: B The Incremented value of myChar2: C

GPT ARAKERE

**Program No.3: Code, execute and debug programs that uses different types of constructors. Identify and resolve issues in the given code snippet.**

class Student

{

String name; int regno;

Student() //Constructor

{ name="Raju";

regno=1234;

}

Student(String n, int r) // parameterized constructor

{ name=n; regno=r;

}

Student(Student s) // copy constructor

{

name=s.name;

regno=s.regno;

} void display()

{

System.out.println(name + "\t" +regno);

}

} class StudentDemo

{ public static void main(String args[])

{

Student s1=new Student();

Student s2=new Student("Ravi",1489); Student s3=new Student(s1); s1.display(); s2.display(); s3.display();

}

}

**Output**:

Raju 1234

Ravi 1489

Raju 1234

Program No.4: Code, execute and debug program to perform autoboxing and unboxing.

Identify and resolve issues in the given code snippet.

class Conversion

{ public static void main(String args[])

{ byte b=10;

short s=20;

int i=30; l

ong l=40;

float f=50.0F;

double =60.0D;

char c='a';

boolean b2=true;

//Autoboxing: Converting primitives into objects

Byte byteobj=b; Short shortobj=s;

Integer intobj=i;

Long longobj=l;

Float floatobj=f;

Doubledoubleobj=d; Character charobj=c;

Boolean boolobj=b2;

//Printing objects

System.out.println("---Printing object values---");

System.out.println("Byte object: "+byteobj);

System.out.println("Short object: "+shortobj);

System.out.println("Integer object: "+intobj);

System.out.println("Long object: "+longobj);

System.out.println("Float object: "+floatobj);

System.out.println("Double object: "+doubleobj);

System.out.println("Character object: "+charobj);

System.out.println("Boolean object: "+boolobj);

Unboxing: Converting Objects to Primitives

byte bytevalue=byteobj;

short shortvalue=shortobj;

int intvalue=intobj;

long longvalue=longobj;

float floatvalue=floatobj;

double doublevalue=doubleobj;

char charvalue=charobj;

boolean boolvalue=boolobj;

//Printing primitives

System.out.println("---Printing primitive values---");

System.out.println("byte value: "+bytevalue);

System.out.println("short value: "+shortvalue);

System.out.println("int value: "+intvalue);

System.out.println("longvalue: "+longvalue);

System.out.println("floatvalue: "+floatvalue);

System.out.println("doublevalue:"+doublevalue);

System.out.println("charvalue: "+charvalue);

System.out.println("boolean value:"+boolvalue);

}

}

**Output**:

---Printing object values--Byte object: 10

Short object: 20

Integer object: 30 Long object:

40

Float object: 50.0

Double object: 60.0 Character object:

a

Boolean object: true

---Printing primitive values--byte value: 10 short value: 20 int value: 30 long value: 40 float value: 50.0 double value: 60.0 char value: a boolean value: true

**Program No.5: Code, execute and debug program to perform evaluation of expression. Identify and resolve issues in the given code snippet.**

import java.util.Scanner;

class ExpressionEvaluation

{ public static void main(String[] args)

{

Scanner input = new Scanner(System.in);

System.out.print("Enter Equation you want to evaluate : ");

String string = input.nextLine(); String a = string.replaceAll(" ",""); if (a.length() < 3)

{

System.out.println( "Please Enter Minimum One Opearator and Two Opearands"); System.exit(0);

} int result = 0; int i = 0;

while(a.charAt(i)!='+' && a.charAt(i)!='-' && a.charAt(i)!='\*' && a.charAt(i)!='/')

{ i++; }

switch (a.charAt(i))

{ case '+' : result = Integer.parseInt(a.substring(0,i))+Integer.parseInt(a.substring(i+1,a.length())); break;

case '-' : result = Integer.parseInt(a.substring(0,i))-Integer.parseInt(a.substring(i+1,a.length()));

break;

case '\*' : result =Integer.parseInt(a.substring(0,i))\*Integer.parseInt(a.substring(i+1,a.length()));

break;

case '/' : result = Integer.parseInt(a.substring(0,i))/Integer.parseInt(a.substring(i+1,a.length())); break;

}

System.out.println(a.substring(0,i) + ' ' + a.charAt(i) + ' ' + a.substring(i+1,a.length())+ " = " + result);

}

}

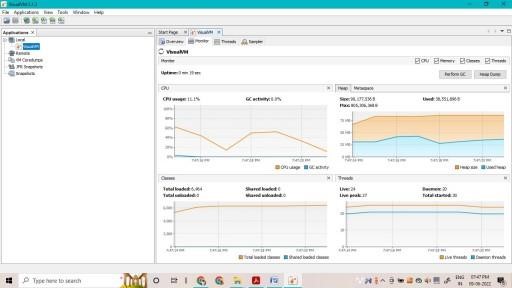
**Output:**

Enter Equation you want to evaluate : 10+30

10 + 30 = 40

**Program No.6: Install memory monitoring tool and observe how JVM allocates Memory.**

* VisualVM provides detailed information about Java applications while they are running on the Java Virutal Machine (JVM).
* VisualVM's graphical user interface enables us to quickly and easily see information about multiple Java applications.
* Installing VisualVM
  + - Download the VisualVM installer from the VisualVM project page.
    - Extract the VisualVM installer to an empty directory on local system.
* Starting VisualVM
  + To start VisualVM on Windows, run the visualvm.exe program that is in the \bin folder under the VisualVM install folder.
  + After opening VisualVM, double click on VisualVM icon which appears on left panel.  Exploring VisualVM:
  + Click on overview tab to know more about JVM arguments, System Properties etc. Click on monitor tab to understand about CPU usage, Heap Space, Classes and Threads. Observe how memory is managed by JVM.
  + Click on Heap Dump(right-side corner) in monitor tab to know about classes, instances, and environment.
  + Click on threads to know how many types threads are running in live and finished threads.  Click on sampler to know CPU samples, Memory Samples.



**Program No :07:-Memory allocation explanation through the programs**

**class** Student

{**int**a;

**int** b;

**public** **void** setData(**int** c,**int** d)

{

a=c;

b=d;

}

**public** **void** showData()

{

System.***out***.println("the value of a="

+a);

System.***out***.println("The value of b="

+b);

}

**public** **static** **void** main(String args[])

{

Student s1 = **new** Student(); Student s2 = **new** Student();

s1.setData(1,2); s2.setData(3,4); s1.showData(); s2.showData();

//Student s3;

//s3=s2;

//s3.showData();

//s2=null;

//s3.showData();

//s3.null;

//s3.showData();

} }

**Output:**

the value of a=1 The value of b=2 the value of a=3 The value of b=4

**Program No.8: Code, execute and debug programs that uses different control statements. Identify and resolve issues in the given code snippet.**

a) Write a Program to check whether the given Integer is Odd or Even using if-else statement. import

java.util.Scanner;

**class** Example

{

**public** **static** **void** main(String args[])

{

**int** age=20;

**if**(age>18)

System.***out***.println("age is greater than 18"); **int** num=13;

**if**(num%2==0)

{

System.***out***.println("\n the

given number is even");

} **else**

{

System.***out***.println("\n the give number is ODD");

}

**int** weight=80; **if**(age>=18)

{

## **if**(weight>50)

{

System.***out***.println("\n you are eligible to donate blood");

}

}

} }

**Output:**

b) Write a program to illustrate switch statement public class

SwitchDemo

{ public static void main(String[] args)

{ int day = 4; switch (day) { case 1:

System.out.println("Monday");

break;

case 2:

System.out.println("Tuesday");

break;

case 3:

System.out.println("Wednesday");

break;

case 4:

System.out.println("Thursday"); break;

case 5:

System.out.println("Friday");

break;

case 6:

System.out.println("Saturday"); break; case 7:

System.out.println("Sunday"); break; }

}

}

**Output:** Thursday

c) Write a Program to Generate n Fibonacci Numbers using for loop. import java.util.Scanner; class Fibonacci

{

public static void main(String[] args)

{

int n, a = 0, b = 0, c = 1;

Scanner s = new Scanner(System.in); System.out.println("Enter value of n:");

n = s.nextInt();

System.out.print("Fibonacci Series:");

for(int i = 1; i <= n; i++)

{ a = b; b = c; c = a + b;

System.out.print(a+" "); }

}

}

**Output:**

Enter value of n:8

Fibonacci Series:0 1 1 2 3 5 8 13

d) Write a Program to Reverse a Number and Check if it is a Palindrome using while loop.

import java.util.Scanner;

class Palindrome

{ public static void main(String args[])

{ int n, m, a = 0,x;

Scanner s = new Scanner(System.in); System.out.println("Enter any number:");

n = s.nextInt(); m = n; while(n > 0)

{ x = n % 10;

a = a \* 10 + x;

n = n / 10;

}

if(a==m)

{

System.out.println("Given number "+m+" is Palindrome");

}

else

{

System.out.println("Given number "+m+" is Not Palindrome");

}

}

}

**Output**:

Enter any number:565

Given number 565 is Palindrome

Enter any number:234

Given number 234 is Not Palindrome

e) Write a Program to Reverse a Number and find the Sum of its Digits using do-while Loop. import

java.util.Scanner;

class DoWhile

{ public static void main(String[] args)

{ int n, a, m = 0, sum = 0;

Scanner s = new Scanner(System.in); System.out.print("Enter any number:");

n = s.nextInt(); do

{

a = n % 10; m = m \* 10

+ a; sum = sum + a; n

= n / 10;

}

while( n > 0);

System.out.println("Reverse:"+m);

System.out.println("Sum of digits:"+sum);

}

}

**Output:**

Enter any number:35

Reverse:53 Sum of digits:8

f) Write a Program to Check whether the given Number is Prime Number. (Use break statement) import

java.util.Scanner;

class CheckPrime

{ public static void main(String args[])

{ int j, x, flag = 1;

System.out.print("Enter any number :"); Scanner s = new Scanner(System.in);

x = s.nextInt();

for( j = 2; j < x; j++)

{

if(x % j == 0)

{ flag = 0; break;

}

} if(flag == 1)

{

System.out.println("The "+x+" is a prime number.");

} else

{

System.out.println("The "+x+" is not a prime number.");

}

}

}

**Output:**

Enter any number :45

The 45 is not a prime number.

Enter any number :23

The 23 is a prime number.

**Program No.9: Code, execute and debug programs that uses encapsulation concept. class**

**Encapsulate**

**class** Student

{

**private** String name;

**public** String getName()

{

**return** name;

}

**public** **void** setName(String name)

{

**this**.name=name;

}

}

**public** **class** Test

{

**public** **static** **void** main(String args[])

{

Student s=**new** Student(); s.setName("vijay");

s.setAge(15);

System.***out***.println("The Name is:"

+s.getName());

System.***out***.println("The age is:"

+s.getAge());

}

}

**Output**:

The Name is:vijay

**Program No.10: Define class & implement like simple calculator and check compliance with SRP.**

class Calculator { // this class is

only responsible for Arithmetic operations

public static int add(int x, int y) { return x + y; }

public static int sub(int x, int y) { return x - y; }

public static int mul(int x, int y) { return x \* y; }

public static int div(int x, int y) {

return x / y; }

}

class ResultPrinter

public static void printResult(int

value)

{

System.out.println("The value is="+value);

}

}

public class calc

{

public static void main(String

args[])

{

int a = Calculator.add(20, 30); ResultPrinter.printResult(a);

int b = Calculator.sub(20, 30); ResultPrinter.printResult(b);

int c = Calculator.mul(20, 30); ResultPrinter.printResult(c);

int d = Calculator.div(20, 30);

ResultPrinter.printResult(d);

} }

**OUTPUT:**

The value is=119

The value is=-10

The value is=600

The value is=0

**Program No.11: Code, execute and debug programs that uses array concept.** a) one dim array

class OneDimArray

{ public static void main(String args[])

{ int a[]=new int[5];//declaration and instantiation a[0]=10;//initialization

a[1]=20; a[2]=70; a[3]=40; a[4]=50;

//traversing array

System.out.println("Elements of array are");

for(int i=0;i<a.length;i++) //length is the property of array

System.out.println(a[i]); }

}

**Output**:

Elements of array are 10

20

70

40

50

b) Java Program to illustrate the use of multidimensional array

class MultiDimArray

{ public static void main(String args[])

{ int arr[][]={{1,2,3},{2,4,5},{4,4,5}}; //declaring and initializing 2D array //printing 2D array

System.out.println("Elements of 2D array are");

for(int i=0;i<3;i++)

{ for(int j=0;j<3;j++)

{

System.out.println(arr[i][j]+" ");

}

System.out.println();

}

} }

**Output:**

Elements of 2D array are 1

2

3

2

4

5

4

4

5

# Program No.12: Code, execute and debug programs to perform string

**manipulation.**

import java.lang.String;

class StringDemo

{

public static void main(String arg[])

{

String s1 = new String("gpt athani"); String s2 =

"GPT ATHANI";

System.out.println("The string s1 is : " + s1);

System.out.println("The string s2 is : " + s2);

System.out.println("Length of the string s1 is : " + s1.length());

System.out.println("Length of the string s2 is : " + s2.length());

System.out.println("The String s1 in Upper Case : " + s1.toUpperCase());

System.out.println("The String s2 in Lower Case : " + s2.toLowerCase());

System.out.println("The first occurrence of a is at the position : “+s1.indexOf('a')); System.out.println("s1 equals to s2 : " + s1.equals(s2));

System.out.println("s1 equals ignore case to s2 : " + s1.equalsIgnoreCase(s2));

System.out.println("Character at an index of 6 is :" + s1.charAt(6));

String s3 = s1.substring(4, 8);

System.out.println("Extracted substring is :" + s3);

System.out.println("After Replacing a with b in s1 : "+ s1.replace('a', 'b'));

System.out.println("After string concat :" + s1.concat(" Karnataka")); String s4 = " This is a book "; //White space before This word.

System.out.println("The string s4 is :" + s4);

System.out.println("After string trim :" +s4.trim());

int result = s1.compareTo(s2);

System.out.println("After compareTo");

if (result == 0)

System.out.println(s1 + " is equal to " + s2); else if (result > 0)

System.out.println(s1 + " is greater than " + s2); else

System.out.println(s1 + " is smaller than " + s2);

}

}

**Output**:

The string s1 is : gpt athani

The string s2 is : GPT ATHANI

Length of the string s1 is : 10

Length of the string s2 is : 10

The String s1 in Upper Case : GPT ATHANI

The String s2 in Lower Case : gpt athani

The first occurrence of a is at the position : 4 s1

equals to s2 : false s1 equals ignore case to s2 :

true Character at an index of 6 is :h

Extracted substring is :atha After Replacing a with b in s1 : gpt bthbni

After string concat :gpt athani Karnataka

The string s4 is : This is a book

**Program No.13:Code, Execute and debug programs that uses inheritance concepts**

**(save with**

**TESTINHERITANCE.java)**

Ex: Single Inheritance  **class** Parent

{

**public** **void** pMethod()

{

System.***out***.println("---Parent method---");

}

}

**class** TESTINHERITANCE

**extends** Parent

{

**public** **void** cMethod()

{

System.***out***.println("---Child method----");

}

**public** **static** **void** main(String

## args[])

{

TESTINHERITANCE cobj = **new** TESTINHERITANCE();

cobj.cMethod(); //method of Child class

cobj.pMethod(); //method of Parent

class

}

}

**Output**:

---Child method----

---Parent method---

Ex2: Multilevel inheritance ( MultiDemo.java)

**class** Animal

{ **void** eat()

{

System.***out***.println("eating...");

}

}

**class** Dog **extends** Animal

{ **void** bark()

{

System.***out***.println("barking...");

}

}

**class** BabyDog **extends** Dog

{

**void** weep()

{

System.***out***.println("weeping...");

}

}

**class** MultiDemo

{

**public** **static** **void** main(String

## args[])

{

BabyDog d=**new** BabyDog();

d.weep();

d.bark();

d.eat();

} } **void** eat()

{

System.***out***.println("eating...");

}

}

**class** Dog **extends** Animal

{ **void** bark()

{

System.***out***.println("barking...");

}

}

**class** BabyDog **extends** Dog

{

**void** weep()

{

System.***out***.println("weeping...");

}

}

**class** MultiDemo

{

**public** **static** **void** main(String

## args[])

{

BabyDog d=**new** BabyDog();

d.weep();

d.bark();

d.eat();

}

}

**Output:**

Weeping…

Barking…

Eating…

**Program No.14: Design a class & implement like file parser and check compliance with OCP.**

class Cuboid

{ public double length; public double breadth; public double height;

}

class Application

{

public double get\_total\_volume(Cuboid geo\_objects[])

{

double vol\_sum = 0;

for (Cuboid geo\_obj : geo\_objects)

{

vol\_sum += geo\_obj.length \* geo\_obj.breadth \* geo\_obj.height;

}

return vol\_sum;

} }

public class OCP

{ public static void main(String args[]) {

Cuboid cb1 = new Cuboid(); cb1.length = 5; cb1.breadth = 10; cb1.height = 15;

Cuboid cb2 = new Cuboid(); cb2.length = 2; cb2.breadth = 4; cb2.height = 6;

Cuboid cb3 = new Cuboid(); cb3.length = 3;

cb3.height = 15;

Cuboid c\_arr[] = new Cuboid[3]; c\_arr[0] = cb1; c\_arr[1] = cb2; c\_arr[2] = cb3;

Application app = new Application ();

double volume = app.get\_total\_volume(c\_arr); System.out.println ("The total volume is " + volume); }

}

**Output**:

The total volume is 1338.0

**Program No.15: Code, execute and debug programs that uses a. static binding**

b. dynamic binding

a) Static binding

class Dog

{

private void eat()

{

System.out.println("Dog is eating...");

}

public static void main(String args[])

{

Dog d1=new Dog();

d1.eat();

}

}

**Output**:

Dog is eating...

b) Dynamic binding

class Animal

{ void eat()

{

System.out.println("animal is eating..."); }

}

class Dog1 extends Animal

{

void eat()

{

System.out.println("dog is eating...");

} public static void main(String args[])

{

Animal a=new Dog1();

a.eat();

}

}

**Output:**

Dog is eating...

**Program No.16: Code, execute and debug program that uses abstract class to achieve abstraction** **(Demoabstract.java) abstract** **class** Demo

{ **abstract** **void** display();

} **public** **class** Demoabstract **extends** Demo

{

**void** display()

{

System.***out***.println("Abstract method called.");

}

**public** **static** **void** main(String[] args)

{

Demoabstract obj = **new**

Demoabstract ();

//invoking abstract method

obj.display();

}

} Output:

---Abstract method called.---

**Program No.17: Code, execute and debug program that uses interface to achieve abstraction.**(car.java)

**interface** CarStart

{

**void** start();

}

**interface** CarStop

{

**void** stop();

}

**public** **class** Car **implements**

CarStart, CarStop

{

**public** **void** start()

{

System.***out***.println("The car engine has been started.");

}

**public** **void** stop()

{

System.***out***.println("The car engine has been stopped.");

}

**public** **static** **void** main(String

## args[])

{

Car c = **new** Car(); c.start();

System.***out***.println(); c.stop();

}

}

Output:

**The car engine has been started.**

**The car engine has been stopped.**

**Program No.18: Code, execute and debug program to read the content of the file and write the content to another file.**

(First create one text file- inputFile.txt and another text file outputFile.txt in C:drive\test folder )

import java.io.File; import java.io.FileInputStream;

import java.io.FileWriter;

import java.io.IOException; import

java.util.Scanner;

class CopyContent

{ public static void main(String[] args) throws IOException {

File file = new File("C:\\test\\inputFile.txt");

FileInputStream inputStream = 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:\\test\\outputFile.txt"); FileWriter writer = new FileWriter(dest); writer.write(buffer.toString()); writer.close(); System.out.println("File copied successfully......."); }

}

Output:

Contents of the file:

Welcome to GPT Athani This is example for checked exceptions.

It uses throws keyword.

Welcome to CS dept File copied successfully.......

**b. read the content of the file and write the content to another file** import java.io.\*; class filereader { public static void main(String arg[]) throws IOException {

int i = 0;

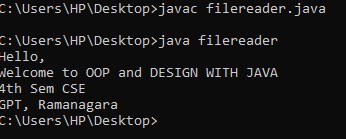
FileReader fr = new

FileReader("C:/Users/HP/Desktop/b.t

xt");

while ((i = fr.read()) != -1) System.out.print((char) i); fr.close();

} }



**Program No.19: Code, execute and debug program that handles checked and unchecked exceptions**

.

import java.io.\*;

class Checked

{

public static void main(String[] args) throws IOException

{

FileReader file = new FileReader("C:\\test\\gpta.txt"); BufferedReader fileInput = new BufferedReader(file);

for (int counter = 0; counter < 3; counter++)

System.out.println(fileInput.readLine());

fileInput.close(); }

}

**Output**:

Welcome to GPT Athani

This is example for checked exceptions. It uses throws keyword.

a) Unchecked Exceptions:

class Unchecked

{ public static void main(String args[])

{

int x = 0;

int y = 10;

int z = y / x;

}

}

Output:



**Program No.20: Code, execute and debug program to illustrate throwing our own exceptions or user defined exceptions.**

import java.lang.Exception;

class MyException extends Exception

{

MyException(String message)

{ super(message);

}

}

class TestMyException

{ public static void main(String args[])

{ int x=5,y=1000; try

{ float z=(float) x/(float) y;

if(z < 0.01)

{ throw new MyException(“Number is too small”);

}

}

catch(MyException e) {

System.out.println(“Caught my exception”);

System.out.println(e.getMessage()); } finally

{

System.out.println(“I am always here”); }

}

}

**Output:**

Caught my exception

Number is too small

I am always here

**Program No.21: Design an interface & implement it like one that builds different types of toys and check compliance with ISP.**

interface Toy

{ void price(doublep);

void color(String c);

}

interface Move

{

void move();

}

interface flying

{

void fly();

}

class car implements Toy,Move

{

double p; String c;

public void price(double p)

{

this.p=p;

System.out.println("The car Toy"); System.out.println("price =" +p);

}

public void color(String c)

{

this.c=c; System.out.println("color =" +c);

}

public void move()

{

System.out.println("start to move forward and backward ");

}

}

class plane implements Toy,flying

{

double p; String c;

public void price(double p)

{ this.p=p;

System.out.println("The Plane Toy");

System.out.println("price =" +p);

}

public void color(String c)

{

this.c=c; System.out.println("color ="

+c);

}

public void move()

{

System.out.println("plane can move

forward and backward ");

}

public void fly()

{

System.out.println("starts flying");

}

}

public class Toyb1

{

public static void main(String args[])

{

car c=new car();

c.price(12.0);

c.color("red");

c.move();

plane p=new plane(); p.price(15.0);

p.color("yellow");

p.move();

p.fly();

}

}

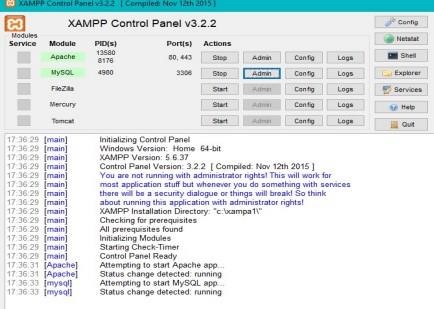
**Output:**

The car Toy price =12.0 color =red start to move forward and backward The Plane Toy

**Program No.22: Code, execute and debug programs to connect to database through JDBC and perform basic DB operations.**

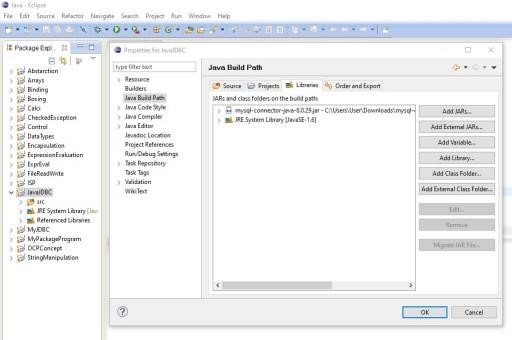
Step 1: In addition to JDK and Eclipse environment, install Xampp software for Apache server and MySql service.

Step 2: Now open Xampp control panel to start Apache and MySql services as shown below. Then click on MySql-Admin button to open MySql http://localhost/phpmyadmin/ in brower.

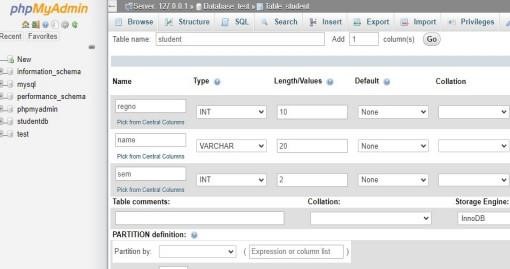


Step 3: To connect MySql databse in Java using Eclipse, follow below steps.  Open Eclipse IDE and create new Java project named JavaJDBC and click finish.  Create a new Java class with DBTest and click on the finish button.

* In order to connect Java program (DBTest.java) with MySQL database, we need to download and include MySQL JDBC driver which is a JAR file, namely mysql-connector-java-8.0.29.jar.
* Now right click on JavaJDBC project to include connector and go to properties. Click on Java build path option-> click on libraries and then click on Add External JARS.  Now select downloaded jar file mysql-connector-java-8.0.29.jar. & click open.  Click on OK and close.



Step 4: Now in browser go to myphpadmin page and create student table in test database with following fields as shown below and click save.



Connecting Java Program with MySQL Database

 After adding jar file, connect the Java program with MySQL Database.

i)Establish a connection using DriverManager.getConnection(String URL) and it returns

a Connection reference.

ii) In String URL parameter write like this :

jdbc:mysql://localhost:3306/test”, “root”, “password” Where,

* jdbc is the API.
* mysql is the database.
* localhost is the name of the server in which MySQL is running.
* 3306 is the port number.
* test is the database name. If the database name is different, then replace this name with the correct database name.
* root is the username of the MySQL database. It is the default username for the MySQL database.
* password is the password that is given while installing the MySQL database.  SQL Exception might occur while connecting to the database, try-catch block must be used.

Step 5: Write below code in DBTest class Eclipse environment.

import java.sql.\*;

public class DBTest {

public static void main(String[] args)

{

String url= "jdbc:mysql://localhost:3306/test"; // table URL

String uname = "root"; // MySQL credentials

String pw = ""; try

{

//Loading MySQL Driver

Class.forName("com.mysql.cj.jdbc.Driver");

// Establishing connection with MySQL

Connection con = DriverManager.getConnection(url,uname,pw);

System.out.println("Java Connection to MySQL Established successfully"); // Creating Statement object for query execution

Statement st=con.createStatement();

// Delete the table student if already present in the test database

String deltbl= "DROP TABLE STUDENT"; st.executeUpdate(deltbl);

// Create a table STUDENT in database test

String qrytbl= "CREATE TABLE STUDENT(regno int,name varchar(30),sem int)"; st.executeUpdate(qrytbl);

// Insert values into the STUDENT table

String qry1="INSERT INTO STUDENT values(2001,'Anand',4)";

st.executeUpdate(qry1);

String qry2="INSERT INTO STUDENT values(2002,'Santosh',4)";

st.executeUpdate(qry2);

String qry3="INSERT INTO STUDENT values(2003,'Ullas',4)";

st.executeUpdate(qry3);

System.out.println("Table Values insertion is successful");

// Query to retrieve values from table

String query= "SELECT \* FROM STUDENT";

ResultSet rs = st.executeQuery(query);//Execute query

while (rs.next())

{

//Retrieve row-wise values of regno, name and sem columns

int regno = rs.getInt("regno");

String name= rs.getString("name");

int sem=rs.getInt("sem");

// Display the result on console

System.out.println(regno + " " + name+ " "+ sem);

} st.close();

// close statement

con.close(); // close

connection

System.out.println("MySQL Connection Closed successfully!");

}

catch(Exception e)

{

System.out.println("Error while executing program:" + e);

}

}

}

**Output**:

