OBJECT ORIENTED PROGRAMMING LAB FILE

Submitted By:

Khushi Wadhawan

500093673

CSF B2

EXPERIMENT 1

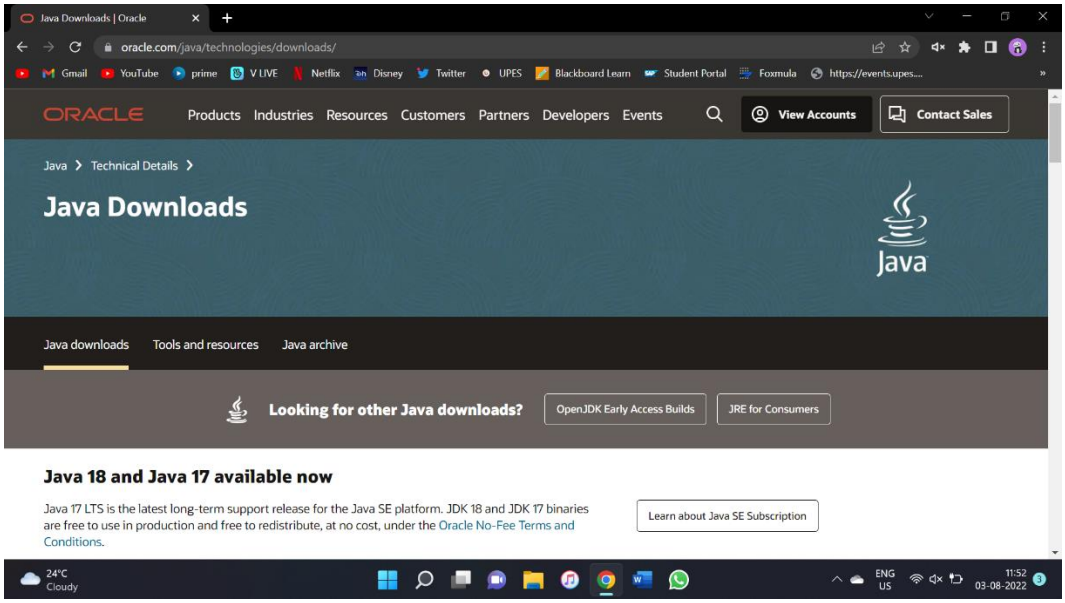
Introduction to java

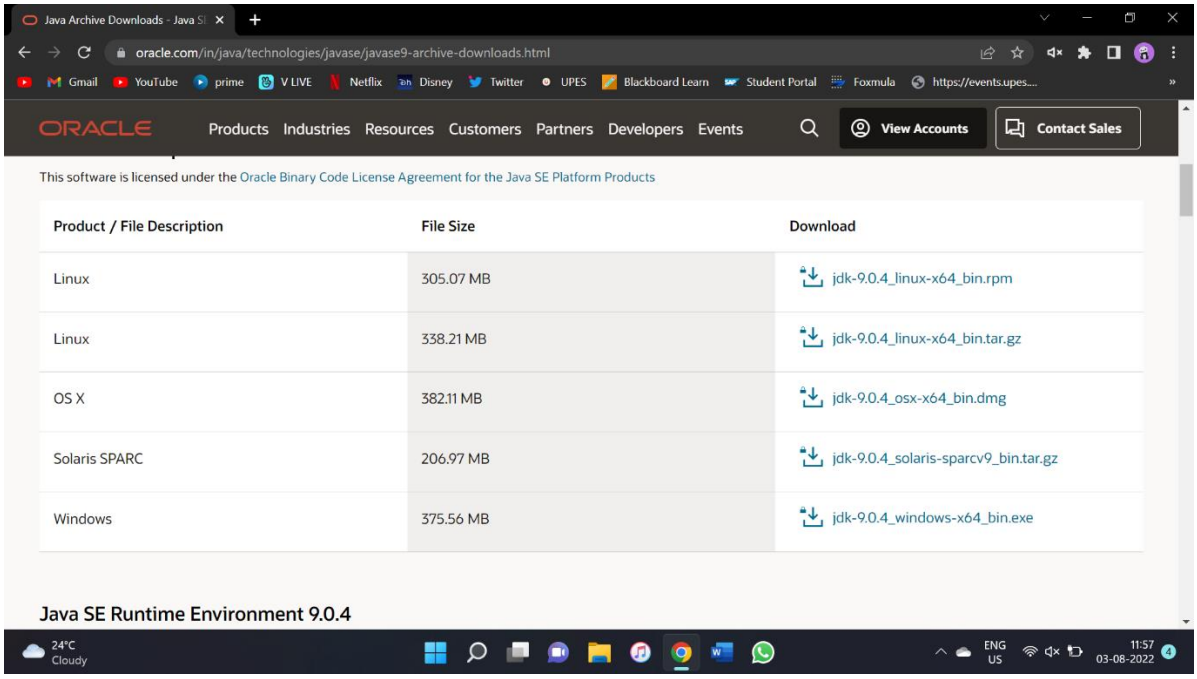
1. Installation of jdk

Open URL:

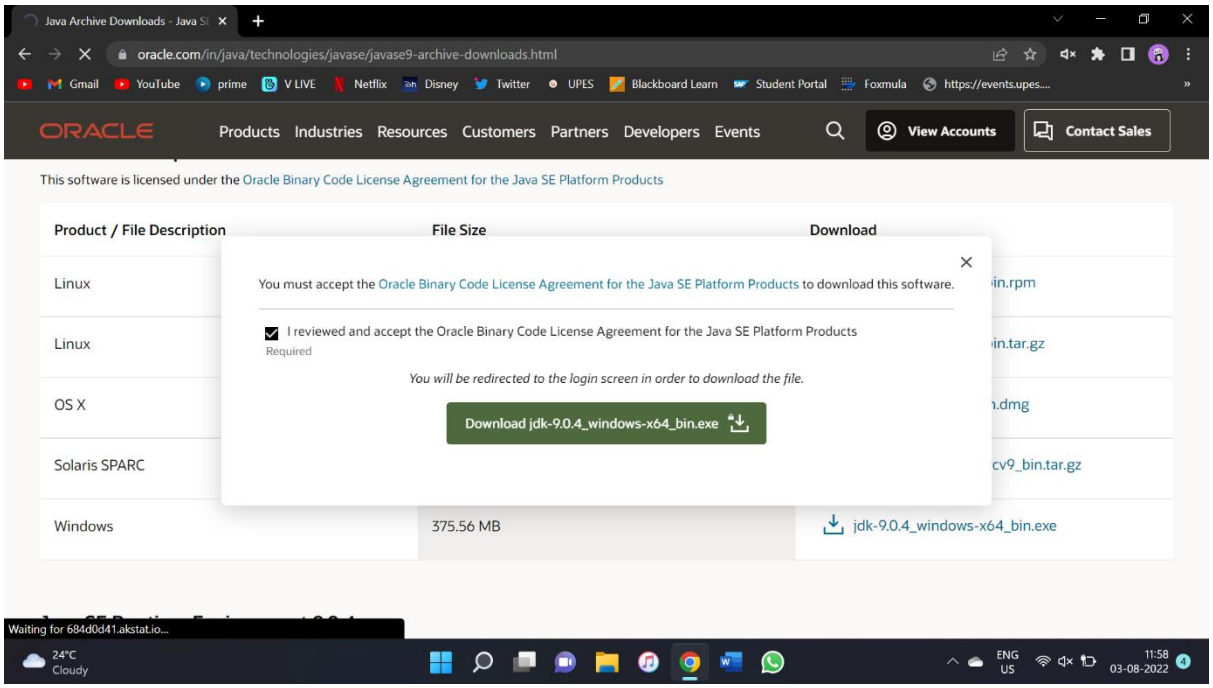
https://www.oracle.com/java/technologies/downloads/

https://www.oracle.com/in/java/technologies/javase/javase9-archive-downloads.html



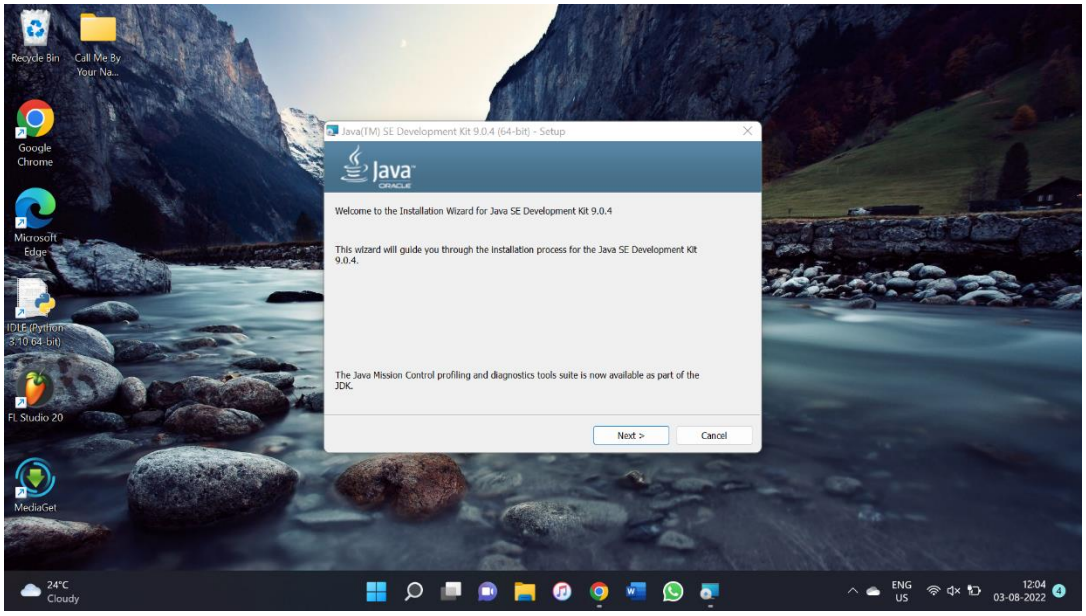


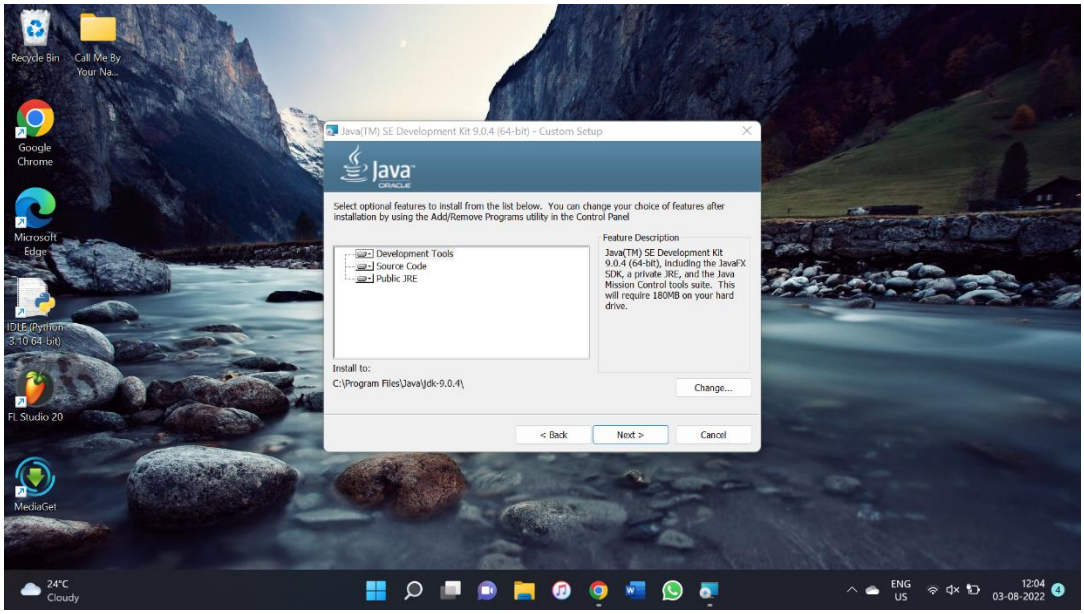
Download the java 9 version for your respective OS

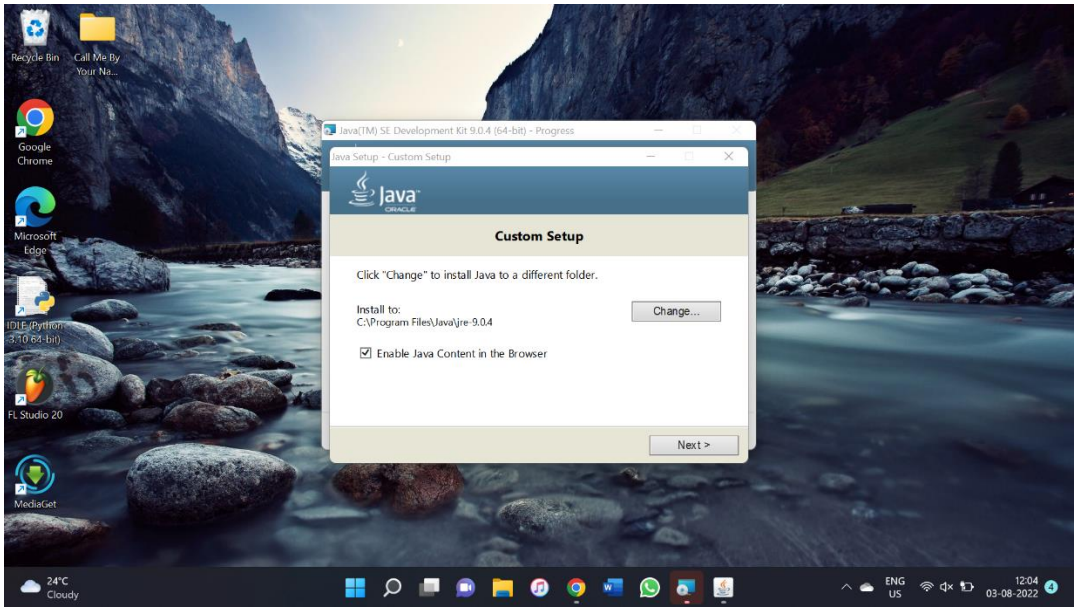


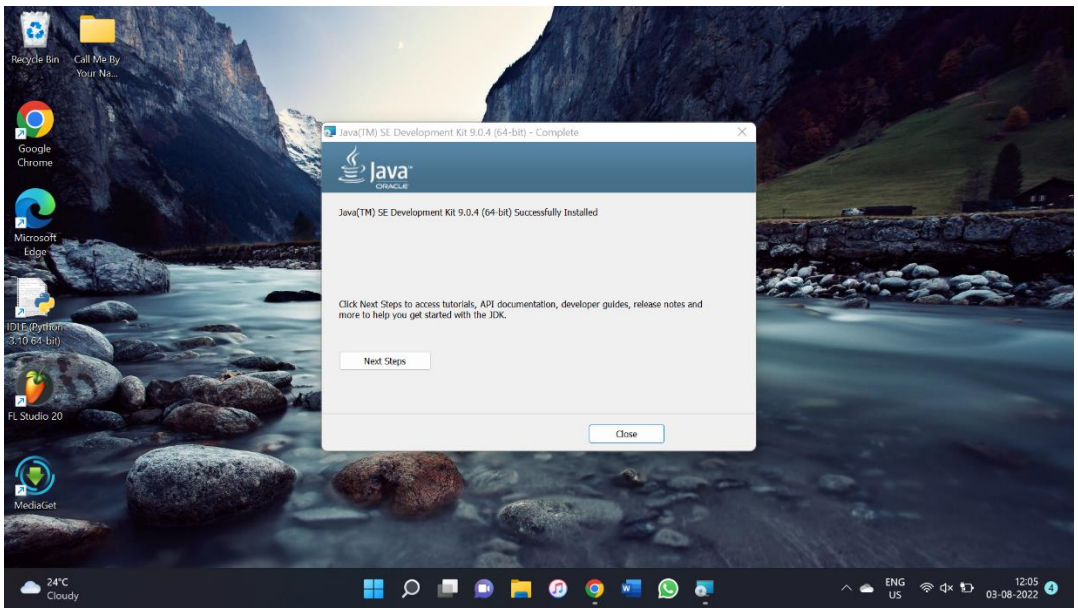
Click on the green download button

Click on Next till installation is complete




Java(TM) SE Development Kit 9.0.4 (64-bit) - Custom Setup
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(TM) SE Development Kit 9.0.4 (64-bit), including the JavaFX SDK, a private JRE, and the Java Mission Control tools suite. This will require 180MB on your hard drive.
Install to:
C:\Program Files\Java\jdk-9.0.4\
Change...
< Back
Next >
Cancel


Java(TM) SE Development Kit 9.0.4 (64-bit) - Progress
Java Setup - Custom Setup
Custom Setup
Click "Change" to install Java to a different folder.
Install to:
C:\Program Files\Java\jre-9.0.4
Change...
Enable Java Content in the Browser
Next >


Java(TM) SE Development Kit 9.0.4 (64-bit) - Complete
Java(TM) SE Development Kit 9.0.4 (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.
Next Steps
Close

2. Setting path variables (Temporary and Permanent)

```
class First
{
	public static void main(String args[])
	{
	System.out.println("hello user");
	}
}
```

javac is the name of java compiler

javac First.java

Setting temporary path:
(C:\Program Files\Java\jdk 9.04\bin)
F:\java>set path=C:\Program Files\Java\jdk 9.04\bin
F:\java>echo %path%

Setting permanent path:

- Go to MyComputer properties
- Go to Advanced tab
- Click on environment variables
- New tab of user variable
- Write path in variable name
- Write path of bin folder in variable value
- Click ok till done

3. Introduction to Java IDE

Java IDE (**Integrated Development Environment**) is a software application that enables users to **write** and **debug** Java programs more easily. Most IDEs have features such as syntax highlighting and code completion that helps users to code more easily. Usually, Java IDEs include a **code editor, a compiler, a debugger**, and **an interpreter** that the developer may access via a single graphical user interface.

Some of the most popularly used IDEs are:

1. Eclipse

- Eclipse provides powerful tools for different software development processes, such as charting, reporting, checking, etc. so that Java developers can build the application as quickly as possible.
- Eclipse can be used on platforms such as MacOS, Linux, Windows, and Solaris.

2. NetBeans

- NetBeans is available for various operating systems, such as Linux, MacOS, Windows, Solaris, etc.

- Although NetBeans is mainly a Java IDE, it has extensions to operate in many other programming languages, such as C, PHP, C++, JavaScript, HTML5, etc.
- NetBeans may be used on different systems such as MacOS, Windows, Solaris and Linux

3. IntelliJ IDEA

- IntelliJ IDEA has several features to make programming easier, like code completion, debugging, XML editing support, code refactoring, code checks, TestNG, unit testing, etc.
- Both versions of IntelliJ IDEA support different programming languages like Java, Kotlin, Groovy, Scala, etc.

4. BlueJ

- The BlueJ main screen displays the current development application's class structure, which is simple to access and change objects.
- BlueJ's basic design is distinct from other IDE's because it was specifically designed to teach OOPS to beginners.

EXPERIMENT 2

1. Write and execute my first java program.

Algorithm:

Step 1: start

Step 2: print Hello User

Step 3: stop

Code:

```
class First
{
	public static void main(String args[])
	{
	System.out.println("Hello User");
	  System.out.println("my name is Khushi Wadhawan - R214220423 ");
	}
}
```

```
D:\java 8>javac First.java

D:\java 8>java First
Hello User
```

2. Print Fibbonacci series.

Algorithm:

Step 1: start

Step 2: Declare variables n1, n2, n3, count , i

Step 3: initialize the variables x = 0 , y = 1 , n = 10

Step 4: print first two terms of series

Step 5: use for loop

Step 6: n3=n1+n2

Step 7: n2=n3

Step 8: n1=n2

Step 9: print n3

Step 10: stop

Code:

```
class Fibo
{
	public static void main(String args[])
	{
	int n1=0,n2=1,n3,i,count=10;
	System.out.println("Khushi Wadhawan  - R214220423 ");
	System.out.print(n1+" "+n2);
	for(i=2;i<count;i++)
	{
	 n3=n1+n2;
	System.out.print(" "+n3);
	n1=n2;
	n2=n3;
	}
	}
}
```

```
0 1 1 2 3 5 8 13 21 34
```

3. Largest of 3 numbers

Algorithm:

Step 1: start

Step 2: declare variables a , b , c

Step 3: initialize variables a= 10 ; b = 20; c= 30

Step 4: check if a>=b and a>=c

Step 5: if true, check if a > c

Step 6: if true, print a is greatest

Else check if b>=a and b>=c

Step 7: if true, print b is greatest

Step 8: Else print c is greatest

Step 9: stop

Code:

```
class Largest
{
	public static void main(String[] args)
	{
	int a=10, b=20, c=30;
	System.out.println("Khushi Wadhawan - R214220423 ");
	if(a>=b && a>=c)
	System.out.println("the largest Number is"+a);
	else if (b>=a && b>=c)
	System.out.println("the largest Number is"+b);
	else
	System.out.println("the largest number is"+c);
	}
}
```

```
the largest number is30

D:\java 8>
```

4. Find sum of all numbers greater than 40 and less than 250 that are divisible by 5

Algorithm:

Step 1: start

Step 2: declare variables i , sum

Step 3: Initialize sum = 0

Step 4: use a for loop

For(i = 41 ; i < 250 ; i++)

Step 5: inside loop check if, i % 5 = = 0

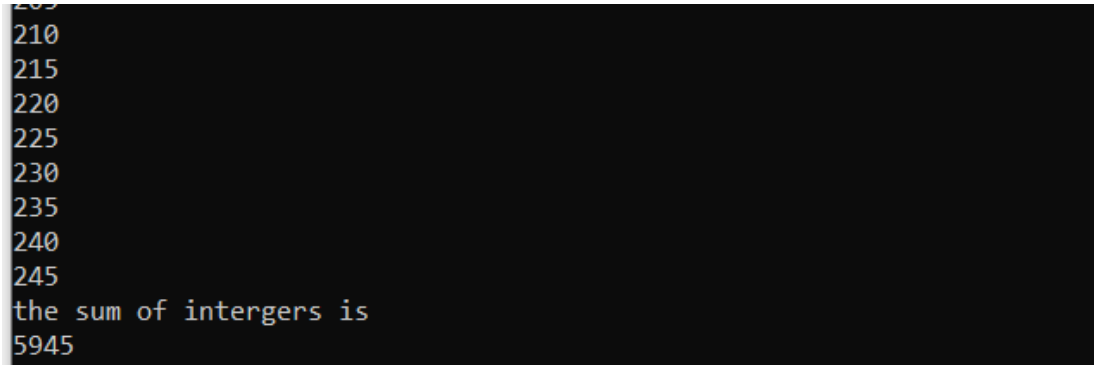
Step 6: if true, sum = sum + i

Step 7: print sum

Step 8: stop

Code:

```
class Sum
{
        public static void main(String arg[])
      {
        int sum=0;
        for(int i=41;i<250;i++)
      {
        if(i%5==0)
      {
        System.out.println(i);
        sum=sum+i;
        }
        }
        System.out.println("the sum of intergers is \n"+sum);
      System.out.println("Khushi Wadhawan - R214220423 ");
      }
}
```



EXPERIMENT 3

Q1. Write a program to take input (a number) of a month (1 - 12) and print its equivalent name of the month. (e.g 1 to Jan, 2 to Feb. 12 to Dec.) Use Scanner class for user input(Hint-use switch case)

CODE:

```
import java.util.Scanner;

public class NumOfMonth {

   public static void main(String[] args) {
    System.out.println("Enter no. 1-12 to find corresponding
month");
    Scanner sc= new Scanner(System.in);
    int num= sc.nextInt();

    switch (num)
    {
    case 1:
        System.out.println ("The name of month number 1 is
January");
        break;
    case 2:
        System.out.println ("The name of month number 2 is
February");
        break;
    case 3:
        System.out.println ("The name of month number 3 is March");
        break;
    case 4:
        System.out.println ("The name of month number 4 is April");
        break;
    case 5:
        System.out.println ("The name of month number 5 is May");
        break;
    case 6:
        System.out.println ("The name of month number 6 is June");
        break;
    case 7:
        System.out.println ("The name of month number 7 is July");
        break;
    case 8:
        System.out.println ("The name of month number 8 is August");
        break;
    case 9:
        System.out.println ("The name of month number 9 is
September");
        break;
```

```
    case 10:
        System.out.println ("The name of month number 10 is
October");
        break;
    case 11:
        System.out.println ("The name of month number 11 is
November");
        break;
    case 12:
        System.out.println ("The name of month number 12 is
December");
        break;
        default:
            System.out.println ("You have entered an invalid
number");
        }
    } // main method
}
```

OUTPUT:

```
Khushi wadhawan, 500093673, R2142210423
Enter no. 1-12 to find corresponding month
4
The name of month number 4 is April
```

Q2. Write a program to add two numbers using command line arguments.

CODE:

```
public class SumOfNumbers4
{
public static void main(String args[])
{
int x = Integer.parseInt(args[0]); //first arguments
int y = Integer.parseInt(args[1]); //second arguments
int sum = x + y;
System.out.println("The sum of x and y is: " +sum);
}
}
```

Q3. Write a program to implement a command line calculator.

CODE:

```
public class Calculator{
public static void main(String[] args){

int numOne = args[0];
int numTwo = args[1];

add(numOne, numTwo) {
    result = numOne + numTwo;
    System.out.println("{numOne} + {numTwo} = {result}");
}

subtract(numOne, numTwo) {
    result = numOne - numTwo;
    System.out.println("{numOne} - {numTwo} = {result}");
}

multiply(numOne, numTwo) {
    result = numOne * numTwo;
    System.out.println("{numOne} * {numTwo} = {result}");
}

divide(numOne, numTwo) {
    result = numOne / numTwo;
    System.out.println("{numOne} / {numTwo} = {result}");
}
}
}
```

Q4. Write a program that takes students' info from the user (using Scanner class) and displaysoutput on a black screen. Use appropriate mutator and accessor. setInfo(),getInfo()

CODE:

EXPERIMENT 4

1. **Write a JAVA program to implement class mechanism. – Create a class, define data members, constructor, methods (setData(args),getData()) and invoke them inside main method.**

   ALGORITHM:

   Step 1: create a class
   Step 2: define roll name marks
   Step 3: make a default constructor
   Step 4: define a method setdata() and getData() to return the result of calculations.
   Step 5: inside the main method make an object of class and pass the vales.
   Step 6: call the setData and getData method
   Step 7: print the output we get from the getData method.
   Step 8: Stop

   CODE:

```
class Student
{
    int roll;
    String name;
    double marks;
    void setData(int a, String b, double c)
    {
        roll=a;
        name=b;
        marks=c;
    }
    void getData()
    {
        System.out.println("Name:"+name);
        System.out.println("Roll no:"+roll);
        System.out.println("Marks:"+marks);
    }

}
class Data{
public static void main (String[] args)
{
    System.out.println();
    Student S1=new Student();
    S1.setData(18,"Khushi",100);
    S1.getData();
}
}
```

2. **Write a JAVA program to implement compile time polymorphism.**

   ALGORITHM:

   Step 1: Start
   Step 2: make a class Poly

Step 3: make default functions for calculations
Step 4: print output.
Step 5: Stop

CODE:

```
class Prog
{
     int add(int n1, int n2)
     {
          return n1+n2;
     }
     int  add(int n1, int n2, int n3)
     {
          return n1+n2+n3;
     }
}
public class Poly
{
   public static void main(String args[])
   {
      Prog obj = new Prog();
      System.out.println("Khushi Wadhawan - R214220423 ");
        System.out.println(obj.add(50, 60));
        System.out.println(obj.add(50, 60, 70));
   }
}
```

3. **Write a JAVA program to implement type promotion in method overloading.**

ALGORITHM:

Step 1: start
Step 2: create a class
Step 3: create a method show() with parameters int n
Step 4: create a method show() with parameters String n
Step 5: in the main method make the object of class
Step 6: call the method show and pass the value that you want.
Step 7: Stop

CODE:

```
class Promotion{

    void show(int n){
        System.out.println("int method");
    }

    void show(String n){
        System.out.println("String method");
    }
    public static void main(String[] args) {
        Promotion t  = new Promotion();
        System.out.println("Khushi Wadhawan - R214220423 ");
        t.show('n');
    }
}
```

```
int method

D:\java 8>
```

4. Write a JAVA program to implement constructor overloading.

ALGORITHM:

Step 1: Start
Step 2: make a class Box
Step 3: define the field variables
Step 4: make default constructors of the class box having different parameter
Step 5: make methods for calculating area and volume.
Step 6: in the main method we make two objects of the class box and pass the values for the default constructor.
Step 7: call the methods area and volume and print the output.
Step 8: Stop

CODE:

```
public class Constructor{
    public static void main(String[] args) {
        Box B1 = new Box(20 , 40 , 60);
        Box B2 = new Box(20 , 40);

        int  ar = B2.area();
        int vol = B1.vol();
        System.out.println("Khushi Wadhawan - R214220423 ");
        System.out.println("Area = " + ar);
        System.out.println("Volume = " + vol);
    }
}

class Box {

    int len, bre , hei;
```

```
    Box(int a, int b, int c) {
        len = a;
        bre = b;
        hei = c;
    }

    Box(int a, int b){
        len = a;
        bre= b;
    }

    int area(){
       int ar = len * bre;
       return ar;
    }

    int vol(){
        int vol = len * bre * hei;
        return vol;
    }

}
```

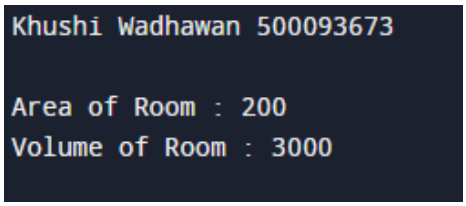
```
Area = 800
Volume = 48000

D:\java 8>
```

EXPERIMENT 5

1. Write a Java program to show that a private member of a super class cannot be accessed from derived classes.

CODE:

```
class room
{
private int l,b;
room(int x,int y)
{ l=x; b=y;}
int area()
{return(l*b);}
}
class class_room extends room
{
int h;
class_room(int x,int y,int z)
{
super(x,y);
h=z;
}
int volume()
{
return(area()*h);
}
}
class s04_01
{
public static void main(String args[])
{
class_room cr=new class_room(10,20,15);
int a1=cr.area();
int v1=cr.volume();

System.out.println(" Khushi Wadhawan 500093673");
System.out.println("Area of Room : "+a1);
System.out.println("Volume of Room : "+v1);
}
}
```



2. Write a program in Java to create a Player class. Inherit the classes Cricket _Player,Football _Player, and Hockey_ Player from Player class.

CODE:

```
class Player
{
  System.out.println(" Khushi Wadhawan 500093673");
  String name;
  int age;
  Player(String n,int a)
  { name=n; age=a; }
  void show()
  {
     System.out.println("Player name: "+name);
     System.out.println("Age: "+age);
  }
}
class cricket_player extends Player
{
  String type;
  cricket_player(String n,String t,int a)
  {
     super(n,a);
     type=t;
  }
  public void show()
  {
     super.show();
     System.out.println("Player type : "+type);
  }
}
class football_player extends Player
  {
     String type;
     football_player(String n,String t,int a)
  {
     super(n,a);
     type=t;
  }
   public void show()
   {
     super.show();
     System.out.println("Player type : "+type);
   }
}
class hockey_player extends Player
{
   String type;
   hockey_player(String n,String t,int a)
   {
      super(n,a);
      type=t;
   }
   public void show()
   {
```

```
      super.show();
      System.out.println("Player type : "+type);
   }
}
class Demo
{
   public static void main(String args[])
   {
     cricket_player c=new cricket_player("Ankit","Cricket",20);
     football_player f=new football_player("Namrata","Football",21);
     hockey_player h=new hockey_player("Aditya","Hockey",22);
     c.show();
     f.show();
     h.show();
   }
}
```

```
Khushi Wadhawan 500093673
Player name: Ankit
Age: 20
Player type : CricketPlayer name: Namrata
Age: 21
Player type : Football
Player name: Aditya
Age: 22
Player type : Hockey
```

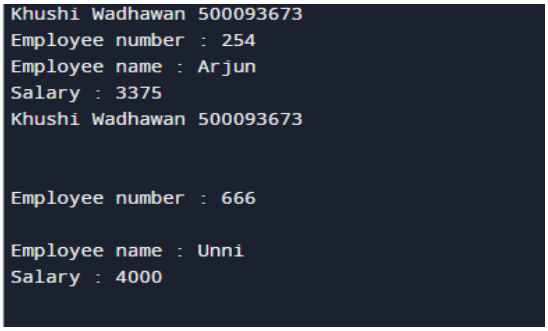
3. Write a class Worker and derive classes DailyWorker and SalariedWorker from it. Everyworker has a name and a salary rate. Write the method ComPay (int hours) to compute theweekly pay of every worker. A Daily Worker is paid on the basis of the number of dayshe/she works. The Salaried Worker gets paid the wage for 40 hours a week no matter whatthe actual hours are. Test this program to calculate the pay of workers. You are expected touse the concept of polymorphism to write this program.

CODE:

```
class worker

{

String name;

int empno;

worker(int no,String n)

{ empno=no; name=n; }

void show()

{
```

```
System.out.println(“Khushi Wadhawan 500093673\n”);
System.out.println("\nEmployee number : "+empno);
System.out.println("\nEmployee name : "+name);
}
}
class dailyworker extends worker
{
int rate;
dailyworker(int no,String n,int r)
{
super(no,n);
rate=r;
}
void compay(int h)
{
show();
System.out.println("Salary : "+rate*h);
}
}
class salariedworker extends worker
{
int rate;
salariedworker(int no,String n,int r)
{
super(no,n);
rate=r;
}
int hour=40;
void compay()
{
show();
System.out.println("Salary : "+rate*hour);
```

```
}
}
//--------- main -----------
class s04_03
{
public static void main(String args[])
{
dailyworker d=new dailyworker(254,"Arjun",75);
salariedworker s=new salariedworker(666,"Unni",100);
d.compay(45);
s.compay();
}
}
```



4. Design a class employee of an organization. An employee has a name, empid, and salary.Write the default constructor, a constructor with parameters (name, empid, and salary) andmethods to return name and salary. Also, write a method increaseSalary that raises theemployee's salary by a certain user-specified percentage. Derive a subclass Manager from the employee. Supply a test program that uses these classes and methods.

CODE:

```
package JavaTpoint.JavaObjectToJSON;
class EmployeeDetails {
int emp_id, salary;
String name, address, department, email;
public int getEmp_id() {
    return emp_id;
}
public void setEmp_id(int emp_id) {
    this.emp_id = emp_id;
```

```
}
public int getSalary() {
    return salary;
}
public void setSalary(int salary) {
    this.salary = salary;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public String toString() {
    return "Employee [emp_id = " + emp_id + ", salary = " + salary +
 ", name = " + name]";
    }
}
public class Employee{
    public static void main(String args[]) {
        EmployeeDetails emp = new EmployeeDetails();
        System.out.println(" Khushi Wadhawan 500093673");

        emp.setEmp_id(101);
        emp.setName("Emma Watson");
        emp.setDepartment("IT");
        emp.setSalary(15000);
        System.out.println(emp);

        int sal = emp.getSalary();
        int increment = 0;

        if ((sal >=1000) && (sal <=1500))
        {

            increment += (sal * 2)/100;
            sal = sal+increment;
```

```
            emp.setSalary(sal);
            System.out.println("\n Salary is incremented \n");
            System.out.println(emp);

        }else if ((sal >=1500) && (sal <=20000)){

            increment += (sal * 5)/100;
            sal = sal+increment;

            emp.setSalary(sal);
            System.out.println("\n Salary is incremented \n");
            System.out.println(emp);
        }else {
            System.out.println("\n Salary is not incremented \n");
            System.out.println(emp);
        }
    }
}
```

EXPERIMENT 6

1. **Write a program to create interface A, in this interface we have two method meth1 and meth2. Implements this interface in another class named MyClass.**

ALGORITHM:

Step 1: Start
Step 2: Create interface A
Step 3: interface B extends A
Step 4: MyClass implements B
Step 5: public void meth1()
Step 6: public void meth2()
Step 7: public void meth3()
Step 8: Stop

CODE:

```
interface B extends A
  {
     void meth3();
       }
 class MyClass implements B
   {
      public void meth1()
   {
     System.out.println("Khushi Wadhawan - R214220423 ");
                System.out.println("Implement meth1().");
                }
public void meth2()
   {
       System.out.println ("Implement meth2().");
         }

public void meth3()
   {
       System.out.println ("Implement meth3()." );
         }
      }
class inter
  {
    public static void main(String arg[])
      {
        MyClass ob = new MyClass();
        ob.meth1();
        ob.meth2();
        ob.meth3();
         }
    }
```

```
Implement meth1().
Implement meth2().
Implement meth3().

D:\java 8>
```

2. Implement Multiple and multilevel Inheritance using Interface.

ALGORITHM:

Step 1: Start
Step 2: Create interface one and interface two
Step 3: interface three extends one,two
Step 4: public class inter2 implements three
Step 5: public void meth1()
Step 6: public void meth2()
Step 7: public void meth3()
Step 8: Stop

CODE:

```
interface one
{
 public void method1();
}
interface two
{
public  void method2();
}
interface three extends one,two
{

 void method3();
}


public class inter2 implements three
{
   public void method1()
   {
     System.out.println("FIRST");
   }
   public void method2()
   {
     System.out.println("SECOND");
   }
   public void method3()
   {
     System.out.println("THIRD");
   }

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

```
    {
      System.out.println("Khushi Wadhawan - R214220423 ");
      inter2 a = new inter2();
      a.method1();
      a.method2();
      a.method3();
    }
}
```

```
FIRST
SECOND
THIRD

D:\java 8>
```

3. Write a program to create an Interface having two methods division and modules. Create a class, which overrides these methods.

ALGORITHM:

Step 1: Start
Step 2: Create interface study
Step 3: class stu implements study
Step 4: int div,mod;
Step 5: class inter3
Step 6: stu s=new stu();
Step 7: print output
Step 8: Stop

CODE:

```
interface study
{
void division(int a);
void modules(int b);
}
class stu implements study
{
String name;
int div,mod;
void name(String n)
{ name=n; }
public void division(int a)
{ div=a; }
public void modules(int b)
{ mod=b; }
void disp()
{
System.out.println("Name :"+name);
System.out.println("Division :"+div);
System.out.println("Modules :"+mod);
}
```

```
}

class inter3
{
public static void main(String args[])
{ stu s=new stu();
s.name("Bharat Gautam");
s.division(56);
s.modules(23);
s.disp();
System.out.println("Khushi Wadhawan - R214220423 ");
}
}
```

```
Division :56
Modules :23
```

4. Write a program to create interface named Test. In this interface, the member function is square. Implement this interface in Arithmetic class. Create one new class called ToTestInt. In this class use the object of Arithmetic class.

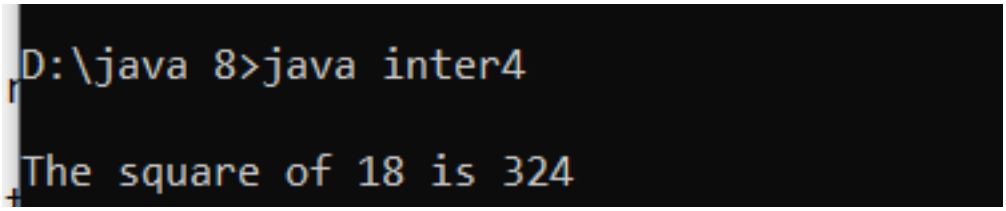
ALGORITHM:

Step 1: Start
Step 2: Create interface test
Step 3: class arithmetic implements test
Step 4: int square()
Step 5: int b, x
Step 6: b = x
Step 7: print output
Step 8: Stop

CODE:

```
interface test
{
int square();
}

class arithmetic implements test
{
      int b;

      arithmetic(int x)
      {
            b = x;
      }

      public int square()
      {
            return (b*b);
```

```
	}

}
class ToTestInt
{
	public int return_ans(int x)
	{
		arithmetic a=new arithmetic(x);
		return a.square();
	}
}
class inter4
{
	public static void main(String []args)
	{
		ToTestInt x= new ToTestInt();
		System.out.println("\nThe square of 18 is
"+x.return_ans(18));
		System.out.println("Khushi Wadhawan - R214220423 ");
	}
}
```



EXPERIMENT 7

1. Write a Java CODE to implement the concept of importing classes from user defined package.

ALGORITHM :-

1. Start
2. Create a package and include a display method in it.
3. Compile the package class
4. Compile the package
5. Make a test class
6. Import the package
7. Call the package method
8. Compile the test class
9. Execute
10. Stop

CODE :-

```
package pack;
public class q1
{
public void message()
{
System.out.println("Khushi Wadhawan - R2142210423 ");
System.out.println("Hello");
}
}
import pack.*;
public class q2
{
public static void main(String args[])
{
pack.c1 obj = new pack.c1();
obj.message();
}
}
```

2. Write a CODE to make a package Balance. This has an Account class with Display_Balance method. Import Balance package in another CODE to access Display_Balance method of Account class.

ALGORITHM :-

1. **Start**

2. **Create a package Balance and include a Display_Balance method in it.**
3. **Compile the package class**
4. **Compile the package**
5. **Make a test2 class**
6. **Import the package**
7. **Call the package method**
8. **Compile the test class**
9. **Execute**
10. **Stop**

CODE:

```
package Balance;
public class Account
{
    public void Display_Balance(int a)
    {
        System.out.println("Khushi Wadhawan - R2142210423 ");
        System.out.println("The account balance is "+a);
    }
}
import Balance.*;
class Exp7
{
    public static void main(String args[])
    {
        Balance.Account obj1=new Balance.Account();
        obj1.Display_Balance(500000);
    }
}
```

3. Write a java CODE that creates a package calculation. Add following classes in it:

a) **Addition**
b) **Subtraction**
c) **Division**
d) **Multiplication**

Write another Test class, import and use the above package.

ALGORITHM :-

1. Start
2. Make the classes addition,multiplication,subtraction and division
3. Ensure all classes fall in the same package
4. Compile each class and then the package
5. Create a class test3 to import the package .
6. Create objects of imported classes
7. Provide arguments
8. Compile test3
9. Execute
10. Stop

CODE:

```
package calc1;
public class add1
{
public void addition(int a , int b)
{
System.out.println("Khushi Wadhawan - R2142210423 ");
System.out.println("Ans:"+(a+b));
}
}
package calc2;
public class sub1
{
public void subtraction(int a , int b)
{
System.out.println("Ans:"+(a-b));
}
}
package calc3;
public class mul1
{
public void multiplication(int a , int b)
{
System.out.println("Ans:"+(a*b));
}
}
package calc4;
public class div1
{
public void division(int a , int b)
{
System.out.println("Ans:"+(a/b));
}
}
```

```
package calc3;
public class mul1
{
public void multiplication(int a , int b)
{

System.out.println("Ans:"+(a*b));
}
}
```

```
package calc4;
public class div1
{
public void division(int a , int b)
{

System.out.println("Ans:"+(a/b));
}
}
```

```
import calc1.*;
import calc2.*;
import calc3.*;
import calc4.*;
class Test6
{
public static void main(String args[])
{
calc1.add1 obj1=new calc1.add1();
calc2.sub1 obj2=new calc2.sub1();
calc3.mul1 obj3=new calc3.mul1();
calc4.div1 obj4=new calc4.div1();
obj1.addition(6,7);
obj2.subtraction(27,13);
obj3.multiplication(13,3);
obj4.division(34,12);
}
}
```

EXPERIMENT 8

Write a program in Java to display the names and roll numbers of students. Initialize respective array variables for 10 students. Handle ArrayIndexOutOfBoundsExeption, so that any such problem doesn't cause illegal termination of program.

Algorithm :-

1. Start
2. Create a class and Declare array and the variable.
3. Input the data using the array in try column
4. If array index is out of the bond it will catch the exception and display it
5. Note the Output
6. Stop

Code:

```
import java.util.*;
import java.io.*;
import java.math.*;
 class Student
{

       public static void main(String []args)
    {
            String name[]= new String[10];
            int marks[] =new int[10];
            int i;
           Scanner sc = new Scanner(System.in);
          try
          {
            for(i=0;i<10;i++)
            {
                  System.out.println("\nEnter name of student
"+(i+1));
                  name[i]=sc.next();
                  System.out.println("\nEnter marks obtained by the
student");
                  marks[i]=sc.nextInt();
            }



       for (i=0;i<10;i++)
            {
                  System.out.println("\nName of student is
"+name[i]);
                  System.out.println("Marks Obtained by the student
are "+marks[i]);

            }
          }
```

```
            catch (ArrayIndexOutOfBoundsException e)
            {
              System.out.println("The array index is out of bound");

            }

    }
}
```

Write a Java program to enable the user to handle any chance of divide by zero exception.

Algorithm :-

1. Start
2. Pick two numbers.
3. One of them is zero.
4. If the division is occurring by 0 Catch the exception
5. Print the output
6. Stop

Code :-

```
class Zero_Division
{
   public static void main (String args[]) {
      int num1 = 15, num2 = 0, result = 0;
      try {
          result = num1/num2;
          System.out.println("The result is:" +result);
      }
      catch (ArithmeticException e) {
         System.out.println ("Can't be divided by Zero " + e);
         System.out.println("Khushi Wadhawan – R2142210423 ");
      }
   }
}
```

Create an exception class, which throws an exception if operand is nonnumeric in calculating modules. (Use command line arguments).

Algorithm :-

1. Start

2. Create a class and Declare array and the variable.

3. Input the data using the array in try column

4. If array index is out of the bond it will catch the exception and display it

5. Note the Output

6. Stop

Code:

```
import java.util.*;
import java.io.*;
import java.math.*;
 class Student
{

       public static void main(String []args)
    {
            String name[]= new String[10];
            int marks[] =new int[10];
            int i;
           Scanner sc = new Scanner(System.in);
          try
          {
            for(i=0;i<10;i++)
            {
                  System.out.println("\nEnter name of student
"+(i+1));
                  name[i]=sc.next();
                  System.out.println("\nEnter marks obtained by the
student");
                  marks[i]=sc.nextInt();
            }



       for (i=0;i<10;i++)
            {
                  System.out.println("\nName of student is
"+name[i]);
                  System.out.println("Marks Obtained by the student
are "+marks[i]);

            }
          }
          catch (ArrayIndexOutOfBoundsException e)
          {
            System.out.println("The array index is out of bound");

          }

    }
}
```

Write a Java program to enable the user to handle any chance of divide by zero exception.

Algorithm :-

1. Start
2. Pick two numbers.
3. One of them is zero.
4. If the division is occurring by 0 Catch the exception
5. Print the output
6. Stop

Code :-

```
class Zero_Division
{
   public static void main (String args[]) {
      int num1 = 15, num2 = 0, result = 0;
      try {
          result = num1/num2;
          System.out.println("The result is:" +result);
      }
      catch (ArithmeticException e) {
         System.out.println ("Can't be divided by Zero " + e);
 System.out.println("Khushi Wadhawan – R2142210423 ");

      }}}
```

3. Create an exception class, which throws an exception if operand is nonnumeric in calculating modules. (Use command line arguments).

Code:

```
class NonNum extends Exception{
NonNum()
{ super("the value is non numeric \n"); }}
class Non_numeric{
public static void main(String ar[]){
int a,b,c=0;
try{
a=Integer.parseInt(ar[0]);
throw new NonNum();
}
catch(NumberFormatException e)
{System.out.println(e);}
catch(NonNum e)
{ System.out.println(e);}
System.out.println("Khushi Wadhawan – R2142210423 ");
}
}
```

Algorithm:

1. Start
2. Use the Exception class and extend it
3. Make a constructor and call it and throw it if a Non Numeric appears

4. Create a new class and Make Try and Exception conditions in it.
5. Note the output
6. Stop

Create an exception class, which throws an exception if operand is nonnumeric in calculating modules. (Use command line arguments).

Algorithm:

1. Start
2. Input the data from user
3. Check if the Entered input in name is letters or there is a number
4. If number is found Throw IO exception
5. Check age if age >0
6. Throw exception
7. Note the output
8. Stop

Code:

```
import java.io.*;
import java.util.*;
class Emp1{
public static void main(String args[]) {
String name;
int age;
System.out.println("-----ENTER EMPLOYEE DETAILS-----");
System.out.println("Enter Name and Age:");
Scanner in=new Scanner(System.in);
try{
if(!(in.nextLine().matches("[a-zA-Z]+")))
{throw new IOException();}
age=in.nextInt();
if(age>50){
System.out.println("Age greater than 50 Exception");
throw new Exception();
}
Emp1 x=new Emp1();
System.out.println("-----Object Created-----");
}
catch(Exception e)
{
System.out.println("Exception");
System.out.println("Khushi Wadhawan – R2142210423 ");
}
}
}


class NonNum extends Exception
{
NonNum()
```

```
{ super("the value is non numeric \n"); }
}
class Non_numeric
{
public static void main(String ar[])
{
int a,b,c=0;
try
{
a=Integer.parseInt(ar[0]);
throw new NonNum();
}

catch(NumberFormatException e)
{System.out.println(e);}
catch(NonNum e)
{ System.out.println(e);}
System.out.println("Khushi Wadhawan – R2142210423 ");
}
}
```

4. Create an exception class, which throws an exception if operand is nonnumeric in calculating modules. (Use command line arguments).

Algorithm :-

1. Start
2. Input the data from user
3. Check if the Entered input in name is letters or there is a number
4. If number is found Throw IO exception
5. Check age if age >0
6. Throw exception
7. Note the output
8. Stop

Code:

```
import java.io.*;
import java.util.*;
class Emp1{
public static void main(String args[]) {
String name;
int age;
System.out.println("-----ENTER EMPLOYEE DETAILS-----");
System.out.println("Enter Name and Age:");
Scanner in=new Scanner(System.in);
try
{
if(!(in.nextLine().matches("[a-zA-Z]+")))
{throw new IOException();}
age=in.nextInt();
if(age>50)
```

```
{
System.out.println("Age greater than 50 Exception");
throw new Exception();
}
Emp1 x=new Emp1();
System.out.println("-----Object Created-----");
}
catch(Exception e)
{
System.out.println("Exception");
System.out.println("Khushi Wadhawan – R2142210423 ");
}
}
}
```

EXPERIMENT 9

Q1. Write a program to implement the concept of multithreading by extending the Thread class.

Algorithm:

Step 1: Start

Step 2: Create a class that extends the Thread class.

Step 3: Declare a function run().

Step 4: Call start() method to start the execution of a thread in main class.

Step 5: Stop

Code:

```
class MyThread1 extends Thread
{
public void run()
{for(int i=1;i<=12;i++)
{System.out.println("Running Thread1:"+i);}
}}
class MyThread2 extends Thread
{public void run()
{for(int i=11;i<=22;i++)
{System.out.println("Running Thread2:"+i);
}}}
class TestThread
{public static void main(String arg[])
{MyThread1 mt1=new MyThread1();
 mt1.start();
MyThread2 mt2=new MyThread2();
 mt2.start();
} }
```

Output:

```
C:\java code>javac TestThread.java

C:\java code>java TestThread
Running Thread1:1
Running Thread1:2
Running Thread2:11
Running Thread2:12
Running Thread2:13
Running Thread1:3
Running Thread1:4
Running Thread1:5
Running Thread1:6
Running Thread1:7
Running Thread1:8
Running Thread1:9
Running Thread1:10
Running Thread1:11
Running Thread2:14
Running Thread2:15
Running Thread2:16
Running Thread2:17
Running Thread2:18
Running Thread2:19
Running Thread2:20
Running Thread2:21
Running Thread2:22
Running Thread1:12
```

Q2. Write a program to implement the concept of multithreading by implementing a Runnable interface.

Algorithm:

Step 1: Start

Step 2: Create a class which implements Runnable interface in java.lang and override run() method.

Step 3: Create a Thread object and call start() method on this object.

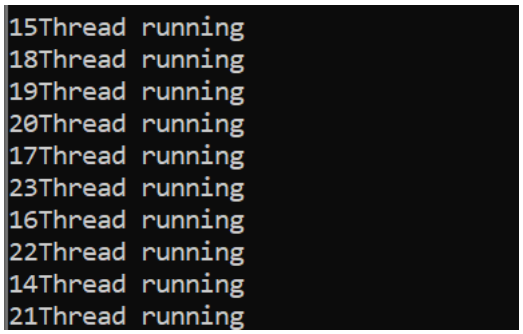
Step 4: Call run() method to start the execution of a thread in main class.

Step 5: Stop

Code:

```
class MulThread implements Runnable {
    public void run()
      {
         try
      {
              System.out.println(Thread.currentThread().getId() +
"Thread running");
      }
         catch (Exception e)
      {
              System.out.println("Exception caught");
      }
      }
      public static void main(String[] args)
      {
         int n = 10;
            System.out.println("Khushi Wadhawan - R2142210423 ");
         for (int i = 0; i < n; i++)
            {
             Thread object = new Thread(new MulThread());
             object.start();
         }}}
```

Output:



Q3. Write a program for generating 2 threads, one for printing even numbers and the other for printing odd numbers.

Code:

```
class MyThread1 extends Thread
{
public void run()
{
for(int i=1;i<=20;i++)
{
  if(i%2==0)
  System.out.println("Running Thread1:"+i);
}
```

```
}
}
class MyThread2 extends Thread
{
public void run()
{
for(int i=11;i<=30;i++)
{
 if(i%2!=0)
 System.out.println("Running Thread2:"+i);
}
}
}
class TestThread1
{
public static void main(String arg[])
{
MyThread1 mt1=new MyThread1();
 mt1.start();
MyThread2 mt2=new MyThread2();
 mt2.start();
}
}
```

Output:

```
C:\java code>java TestThread1
Running Thread1:2
Running Thread2:11
Running Thread1:4
Running Thread2:13
Running Thread1:6
Running Thread2:15
Running Thread1:8
Running Thread2:17
Running Thread2:19
Running Thread1:10
Running Thread2:21
Running Thread1:12
Running Thread2:23
Running Thread1:14
Running Thread2:25
Running Thread1:16
Running Thread2:27
Running Thread1:18
Running Thread2:29
Running Thread1:20
```

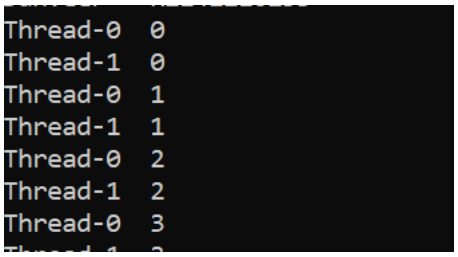
Q4. Write a Java program that implements multithreading among 3 threads. Use sleep() and join() methods and show appropriate output.

Code:

```
import java.lang.*;
class Sleep implements Runnable
{
```

```
    Thread T;
    public void run()
    {
        for (int i = 0; i < 4; i++)
        {
            System.out.println(Thread.currentThread().getName()+ "
" + i);
            try
            {
                Thread.sleep(100);
            }

            catch (Exception e)
            {
                System.out.println(e);
            }
        }
    }
     public static void main(String[] args) throws Exception
    {
System.out.println("Khushi Wadhawan - R2142210423 ");
        Thread T = new Thread(new Sleep());
        T.start();
        Thread t2 = new Thread(new Sleep());
        t2.start();
    }
}
class Join implements Runnable
{
    public void run()
    {
        Thread T = Thread.currentThread();
        System.out.println("Current thread: " + T.getName());
    }
    public static void main(String args[]) throws Exception
    {
System.out.println("Khushi Wadhawan  - R2142210423 ");
        Thread T = new Thread(new Join());
        T.start();
        T.join(500);
        System.out.println("Joining after 500"+" micro seconds:
\n");
        System.out.println("Current thread: " + T.getName());
    }
}
```

Output:



Q5. Write a Java program that shows multithreading between three threads. Set different priorities for each thread and show output.

```
import java.lang.*;
class priority extends Thread
{
    public void run()
      {
            System.out.println("Inside Run Method");
      }
    public static void main(String[] args)
    {
System.out.println("Khushi Wadhawan-R2142210423");


        priority T1 = new priority();
        priority T2 = new priority();
        priority T3 = new priority();
        System.out.println("T1 thread priority : " +
T1.getPriority());
        System.out.println("T2 thread priority : " +
T2.getPriority());
        System.out.println("T3 thread priority : " +
T3.getPriority());
        T1.setPriority(4);
        T2.setPriority(1);
        T3.setPriority(9);
        System.out.println("T1 thread priority : " +
T1.getPriority());
```

```
        System.out.println("T2 thread priority : " +
T2.getPriority());

        System.out.println("T3 thread priority : " +
T3.getPriority());

        System.out.println("Currently Executing Thread : "+
Thread.currentThread().getName());

        System.out.println("Main thread priority : " +
Thread.currentThread().getPriority());

        Thread.currentThread().setPriority(20);

        System.out.println("Main thread priority : " +
Thread.currentThread().getPriority());

    }

}
```

EXPERIMENT 10

Q1.

```
import java.io.*;
import java.lang.String;
import java.util.Scanner;
class StrMethods
{
public static void main(String args[])
{
System.out.println("Khushi - R2142210423 ");
Scanner scan = new Scanner(System.in);
String str1=" ";
String str2=" ";
System.out.println("Enter String 1:");
str1=scan.nextLine();
System.out.println("Enter String 2:");
str2=scan.nextLine();
System.out.println("ConCat:");
System.out.println(str1.concat(str2));
System.out.println("Equals: ");
System.out.println(str1.equals(str2));
System.out.println("Equals Ignore Case:");
System.out.println(str1.equalsIgnoreCase(str2));
System.out.println("UpperCase: ");
System.out.println(str1.toUpperCase());
System.out.println(str2.toUpperCase());
System.out.println("CharAt:");
int pos=0;
System.out.println("Enter Position of word: ");
pos=scan.nextInt();
System.out.println(str1.charAt(pos));
```

```
System.out.println("CompareTo:");
System.out.println(str1.compareTo(str2));
   }
}
```

Q2.

```
import java.io.*;
import java.lang.String;
import java.util.Scanner;
class StrMethods2
{
public static void main(String[] args)
{
System.out.println("Khushi - R2142210423 ");
Scanner scan = new Scanner(System.in);
String str1=" ";
int ch=0;
char choice;
System.out.println("Enter String");
str1=scan.nextLine();
System.out.println("HashCode:");
System.out.println(str1.hashCode());
System.out.println("Trim:");
System.out.println(str1.trim());
System.out.println("Intern:");
System.out.println(str1.intern());
System.out.println("Length:");
System.out.println(str1.length());
System.out.println("Replace:");
System.out.println(str1.replace('K', 'M'));
System.out.println("SubString:");
String subStr = str1.substring(2,4);
```

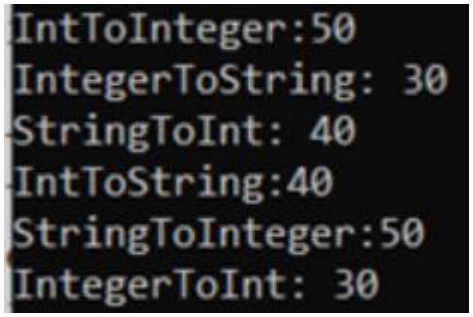
```
System.out.println(subStr);
}
}
```

Q3.

```
class IntToInteger
{
void IntToInteger()
{
int i = 50;
Integer intObj = new Integer(i);
System.out.println("IntToInteger:"+intObj);
}
}
class IntegerToString
{
void IntegerToString()
{
Integer intObj = new Integer(30);
String str = intObj.toString();
System.out.println("IntegerToString: " + str);
}
}
class StringToInt
{
void StringToInt()
{
String s="40";
int i=Integer.parseInt(s);
System.out.println("StringToInt: "+i);
}
```

```
}
class IntToString
{
void IntToString()
{
int i=30;
String s=String.valueOf(i);
System.out.println("IntToString:" + (i+10));
}
}
class StringToInteger
{
void StringToInteger()
{
String s="50";
Integer i=Integer.valueOf(s);
System.out.println("StringToInteger:"+i);
}
}
class IntegerToInt{
    void IntegerToInt()
    {
        Integer intobject = new Integer(30);
int i = intobject.intValue();
System.out.println("IntegerToInt: " + i);
    }
}
public class conversion1
{
public static void main(String args[]) {
System.out.println("Khushi - R2142210423 ");
IntToInteger obj1 = new IntToInteger();
```

```
IntegerToString obj2 = new IntegerToString();
StringToInt obj3 = new StringToInt();
IntToString obj4 = new IntToString();
StringToInteger obj5 = new StringToInteger();
IntegerToInt obj6 = new IntegerToInt();
obj1.IntToInteger();
obj2.IntegerToString();
obj3.StringToInt();
obj4.IntToString();
obj5.StringToInteger();
obj6.IntegerToInt();
}
}
```



Q4.

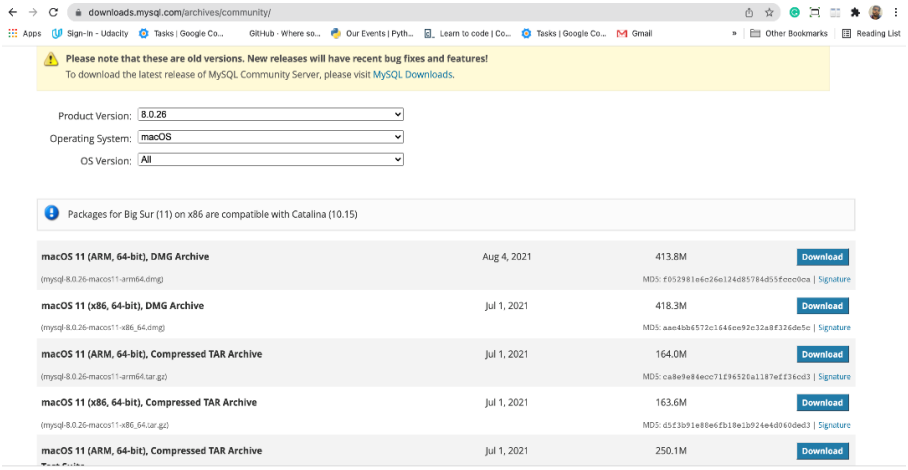
```
import java.util.Scanner;
class convt
{
public static void main(String args[])
{
System.out.println ("Khushi  R2142210423");
Scanner scan = new Scanner(System.in);
float f = 10.0f;
System.out.println("float TO Float");
Float f1 = new Float(f);
System.out.println(f1);
System.out.println("Float To String");
```

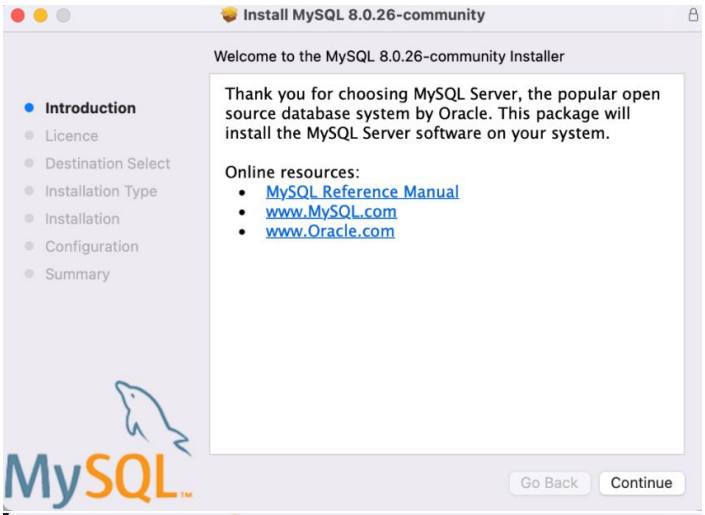
```
String sf = Float.toString(f1);
System.out.println(sf);
System.out.println("String To Float");
System.out.println("enter a string");
String s = scan.next();
Float x=Float.valueOf(s);
System.out.println(x);
System.out.println("float TO String");
String st1 = Float.toString(x);
System.out.println(st1);
System.out.println("String TO Float");
Float in = new Float(st1.valueOf(st1));
System.out.println(in);
System.out.println("Float TO float");
float  no = (float)in;
System.out.println(no);
}
}
```

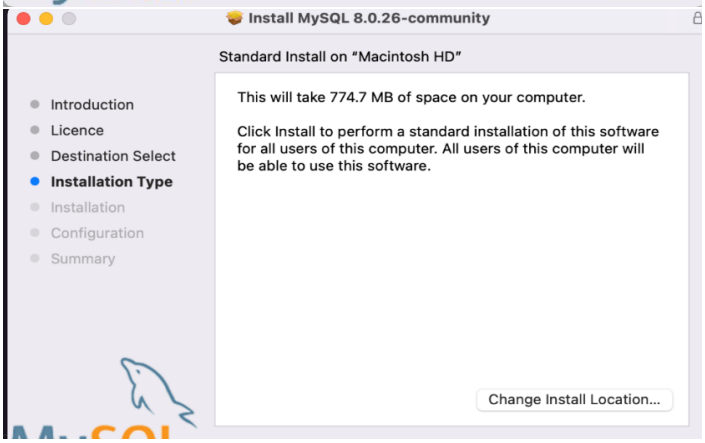
```
float TO Float
10.0
Float To String
10.0
String To Float
enter a string
20
20.0
float TO String
20.0
String TO Float
20.0
Float TO float
20.0
```

EXPERIMENT 11

1.Show the steps of MYSQL/other database server Installation and start the MYSQL or Other Database Server and MYSQL/other client.



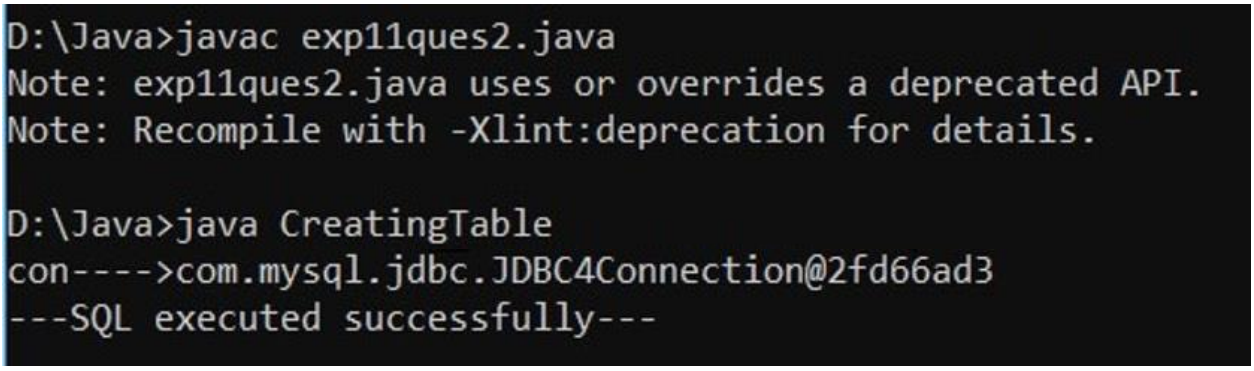




2. Create a database table to store the records of students in college. Use getConnection function to connect the database. The statement object uses executeUpdate function to create a table.

Code:

```
import java.sql.*;

class CreatingTable

{

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

{

String driverClassName = "com.mysql.jdbc.Driver";

String url = "root";

String pwd = "root";

Class.forname(driverClassName).newInstance();
System.out.println("con---->"+con);

Statement stat = con.createStatement();

String sql = "CREATE TABLE employee_table(empID int primary key
auto_increment, name varchar(100),salary float(10,2))";

stat.executeUpdate(sql);

sql="INSERT INTO employee_table(name, salary) VALUE(Devashish
Sharma',900000.00)";

stat.executeUpdate(sql);

stat.close();

con.close();

System.out.println("---SQL executed successfully---");

}

}
```



3. Write a Java code that converts int to Integer, converts Integer to String, converts String to int, converts int to String, converts String to Integer, converts Integer to int.

Code:

```
import java.sql.*;
class InsertingTable {
public static void main(String[] args) throws Exception {
String driverClassName = "com.mysql.jdbc.Driver";
String url="jdbc: mysql://localhost:3307/jdbc";
String user="root";

String pwd="root";
Class.forName (driverClassName) .newInstance();
Connection con = Driver Manager.getConnection (url, user, pwd);
System.out.println("con---->"+con);
Statement stat = con.createStatement();
stat.executeUpdate (sql);
String sql= "INSERT INTO employee_table (name, salary)
VALUE(‘Rahul ', 234000.00)" ;
stat.executeUpdate (sql);
sql = "INSERT INTO employee_table (name, salary) VALUE('Pankaj',
32000.05)";
stat.executeUpdate (sql);
sql = "INSERT INTO employee_table (name, salary) VALUE ('Ram',
458000.05)" ;
stat.executeUpdate (sql);
sql = "INSERT INTO employee_table (name, salary) VALUE ('Sejal',
554000.05)";
stat.executeUpdate (sql);
ResultSet resultset - stat.executeQuery("select from
employee_table");
while (resultset.next())
{
System.out.println(resultset.getInt ("empId") +"
"+resultset.getString("name")+""+resultset.getFloat ("salary")); }
resultset.close();
stat.close();
```

```
con.close();
System.out.println("---SQL executed successfully---"); }
}
```