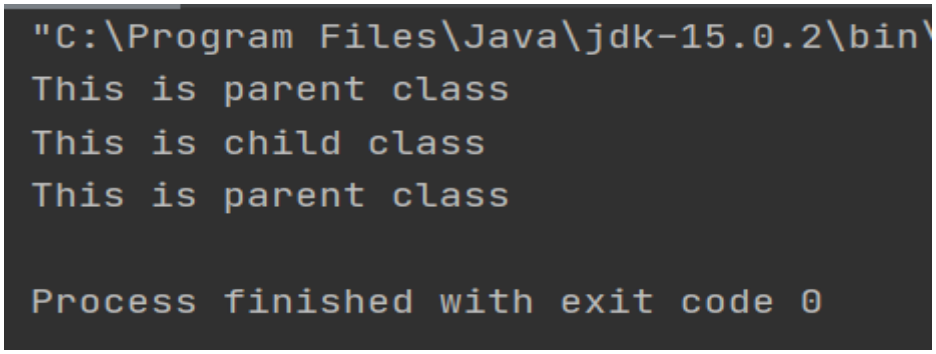
OOPS Lab assignment 2

1. Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call

1 - method of parent class by object of parent class
2 - method of child class by object of child class
3 - method of parent class by object of child class

```
package com.deepak;
import java.util.Scanner;
class parent
{
   void pprint()
   {
      System.out.println("This is parent class");
   }
}
class child extends parent
{
   void cprint()
   {
      System.out.println("This is child class");
   }
}

public class Main {
   public static void main(String[] args) {
      parent a=new parent();
      child b=new child();
      a.pprint();//method of parent called by parent
      b.cprint();//method of child called by child
      b.pprint();//method of parent called by child
   }
}
```



2. In the above example, declare the method of the parent class as private and then repeat the first two operations (You will get error in the third).

```
package com.deepak;
import java.util.Scanner;
class parent
{
   private void pprint()
   {
      System.out.println("This is parent class");
   }
}

class child extends parent
{
   void cprint()
   {
      System.out.println("This is child class");
   }
}

public class Main {
   public static void main(String[] args) {
      parent a=new parent();
      child b=new child();
      a.pprint();//method of parent called by parent
      b.cprint();//method of child called by child
      b.pprint();//method of parent called by child
   }
}
```

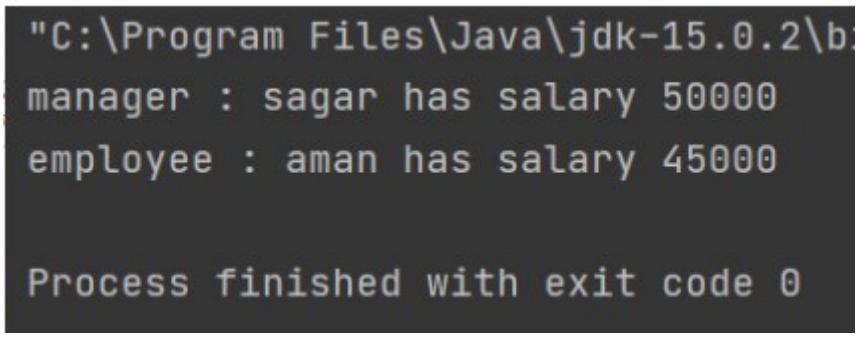
//output gives error as private mathod cannot be accessed bt child class

3. Create a class named 'Member' having the following members: Data members

1 - Name
2 – Age
3 - Phone number
4 – Address
5 - Salary
It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same

```
package com.deepak;
import java.util.Scanner;
class member
{
   int age,salary;
   String phone_no,name,address;
   member (String a,int b, String c,String d, int e )
   {
```

```
        name=a;
        age=b;
        phone_no=c;
        address=d;
        salary=e;
    }
    void printsalary()
    {
        System.out.println(name+" has salary "+salary);
    }
}

class employee extends member
{
    String specialization,department;
    employee(String a,int b, String c,String d, int e, String f, String g )
    {
        super(a,b,c,d,e);
        specialization=f;
        department=g;
    }
    void printsalary()
    {
        System.out.print("employee : ");
        super.printsalary();
    }
}

class manager extends member
{
    String specialization,department;
    manager(String a,int b, String c,String d, int e, String f, String g )
    {
        super(a,b,c,d,e);
        specialization=f;
        department=g;
    }
    void printsalary()
    {
        System.out.print("manager : ");
        super.printsalary();
    }
}
public class Main {
    public static void main(String[] args) {
        manager a=new manager("sagar",25,"899999999","kolkata",50000,"finance","sales execution");
        a.printsalary();
        employee b=new employee("aman",25,"999999999","delhi",45000,"developer","web");
        b.printsalary();
    }
}
```



4. Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'. Print the area and perimeter of a rectangle and a square.

```
package com.deepak;
import java.util.Scanner;

class rectangle
{
   int length,breadth;
   rectangle(int l, int b)
   {
      length=l;
      breadth=b;
   }
   void area()
   {
      System.out.println("area is : "+ length*breadth);
   }
   void perimeter()
   {
      System.out.println("perimeter is : "+ 2*(length+breadth));
   }
}

class square extends rectangle
{
   square(int a)
   {
      super(a,a);
   }
}

public class Main {
   public static void main(String[] args) {
      System.out.println("for rectangle with sides 4 5 ");
     rectangle a=new rectangle(5,4);
     a.area();
     a.perimeter();
      System.out.println("for square with sides 5 5 ");
      square b=new square(5);
```

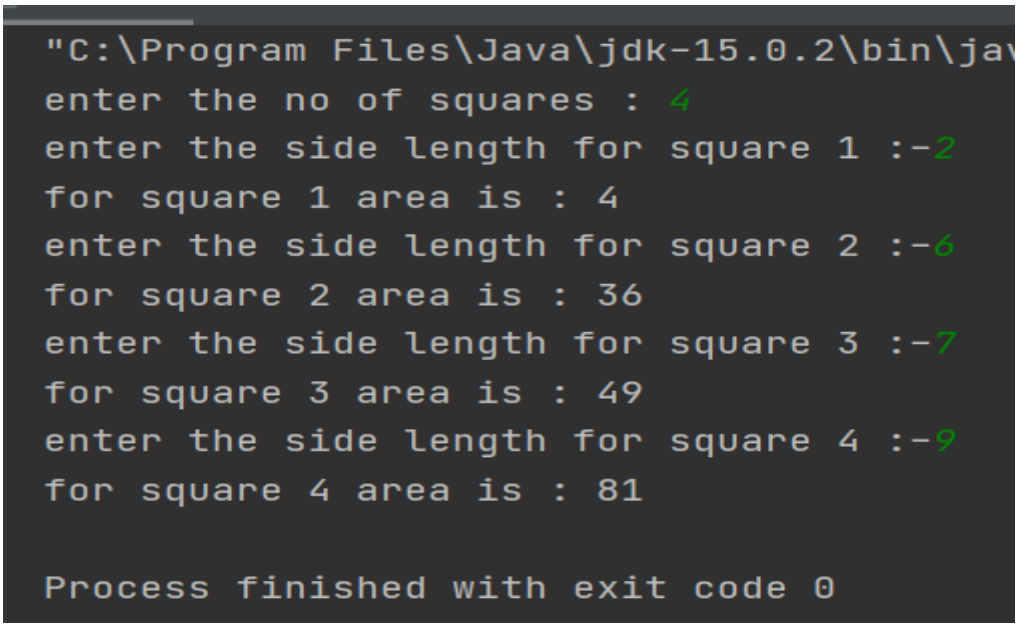
```
    b.area();
    b.perimeter();
  }
}
```

```
"C:\Program Files\Java\jdk-15.0.2\
for rectangle with sides 4 5
area is : 20
perimeter is : 18
for square with sides 5 5
area is : 25
perimeter is : 20

Process finished with exit code 0
```

5. Now repeat the above example to print the area of 10 squares. Hint-Use array of objects

```
package com.deepak;
import java.util.Scanner;

class rectangle
{
  int length,breadth;
  rectangle(int l, int b)
  {
    length=l;
    breadth=b;
  }
  void area()
  {
    System.out.println("area is : "+ length*breadth);
  }
  void perimeter()
  {
    System.out.println("perimeter is : "+ 2*(length+breadth));
  }
}

class square extends rectangle
{
  square(int a)
  {
    super(a,a);
  }
}

public class Main {
  public static void main(String[] args) {
    square a[];
    int x,n;
    Scanner sc=new Scanner(System.in);
```

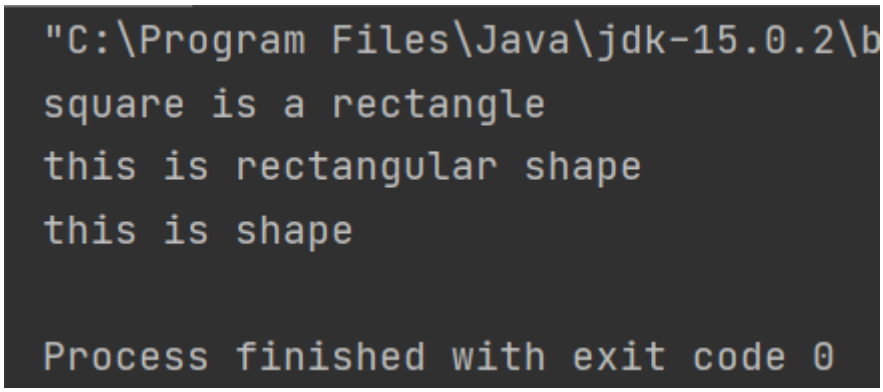
```
        System.out.print("enter the no of squares : ");
        n=sc.nextInt();
        a=new square[n];
        for (int i=0;i<n;i++)
        {
            System.out.print("enter the side length for square "+(i+1)+" :-");
            x=sc.nextInt();
            a[i]=new square(x);
            System.out.print("for square "+ (i+1)+" ");
            a[i].area();
        }
    }
}
```



6. Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class

```
package com.deepak;
import java.util.Scanner;

class shape
{
    String name;
    void sh_print()
    {
        System.out.println("this is shape");
    }
}
class rectangle extends shape
{
    void r_print()
    {
        System.out.println("this is rectangular shape");
    }

}
class circle extends shape
```

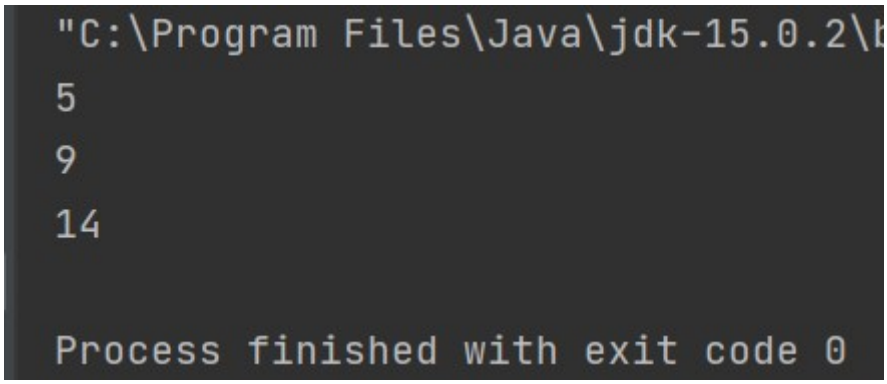
```
{
   void c_print()
   {
      System.out.println("this is circular shape");
   }
}

class square extends rectangle
{
   void s_print()
   {
      System.out.println("square is a rectangle");
   }
}

public class Main {
   public static void main(String[] args) {
      square a=new square();
      a.s_print();
      a.r_print();
      a.sh_print();

   }
}
```



7. Write a Java Program to find the sum of multiple numbers using Method Overloading.

```
package com.deepak;
import java.util.Scanner;

class overload
{
   static int sum(int a, int b)
   {
      return a+b;
   }
   static int sum(int a,int b, int c)
   {
      return a+sum(b,c);
   }
   static int sum(int a, int b, int c, int d)
   {
      return a+sum(b,c,d);
   }
}
public class Main {
```

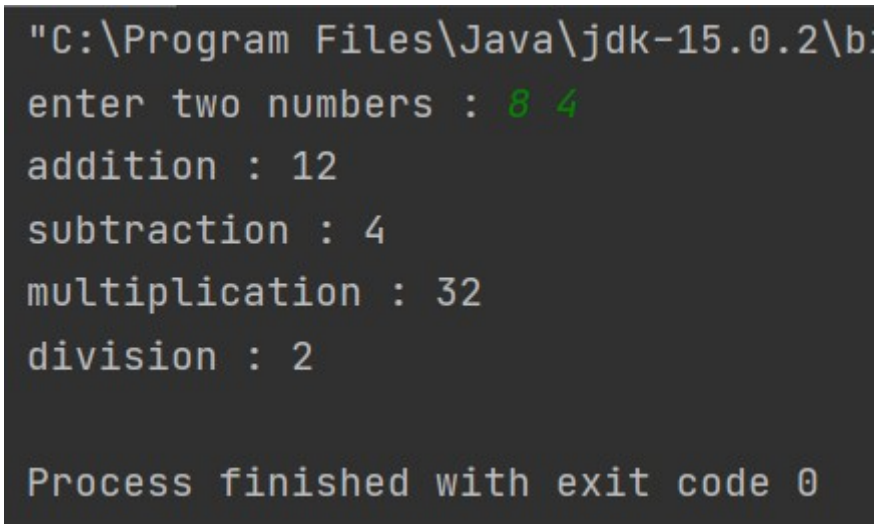
```
    public static void main(String[] args) {
        // here function sum is overloaded to find sum of 2 or 3 or 4 numbers
        int n=overload.sum(2,3);//for 2 no
        System.out.println(n);
        n=overload.sum(2,3,4);//for 3 no
        System.out.println(n);
        n=overload.sum(2,3,4,5);//for 4 no
        System.out.println(n);
    }
}
```



8. Write a program in Java to implement a calculator having four functions such addition, multiplication, division, and subtraction, where the four said functions are defined in four different packages. Inputs are user defined and use the concept of inheritance for the division operation.

```
package com.deepak;
import java.util.Scanner;
import com.add.addition;
import com.subtract.subtraction;
import com.multiply.multiply;
import com.division.division;

//here class calculator is implemented using functions in other packages
class calculator
{
    int add(int a, int b)
    {
        return addition.add(a,b);
    }
    int subtract(int a, int b)
    {
        return subtraction.minus(a,b);
    }
    int multiply(int a,int b)
    {
        return multiply.product(a,b);
    }
    int divide(int a, int b)
    {
        return division.divide(a,b);
    }
}

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

```
        Scanner sc=new Scanner(System.in);
        int a,b;
        calculator cal=new calculator();
        System.out.print("enter two numbers : ");
        a=sc.nextInt();
        b=sc.nextInt();
        System.out.println("addition : "+cal.add(a,b) );
        System.out.println("subtraction : "+cal.subtract(a,b) );
        System.out.println("multiplication : "+cal.multiply(a,b) );
        System.out.println("division : "+cal.divide(a,b) );
    }
}
```



9. Write a java program to throw a custom exception, you are trying to read data from the file. If the file contains more than the specified number (K) characters, k should be taken from the user and k>=100.

```
package com.deepak;
import java.util.Scanner;
import java.io.*;

class myexception extends Exception{
    int length;
    myexception(int a)
    {
        length=a;
    }
    public String  toString(){
        return "the file has more than " + length + " characters";
    }
}

public class Main {
    public static void main(String[] args) throws IOException{
        Scanner sc=new Scanner(System.in);
        System.out.print("enter the value of k: ");
        int k=sc.nextInt(),c=0;
```

```
    FileInputStream f=new FileInputStream("C:\\Users\\Deepak\\Desktop\\sample.txt");
    /* contents of samplt.txt is

    hello this is a sample file to test our program

     */
    try{
       while (f.available()!=0)
       {
          f.read();
          c++;
          if (c>k)
             throw new myexception(k);
       }
       System.out.println("no exception found");
    }
    catch (myexception e)
    {
       System.out.println("exception found " + e);
    }

  }
}
```

with exception

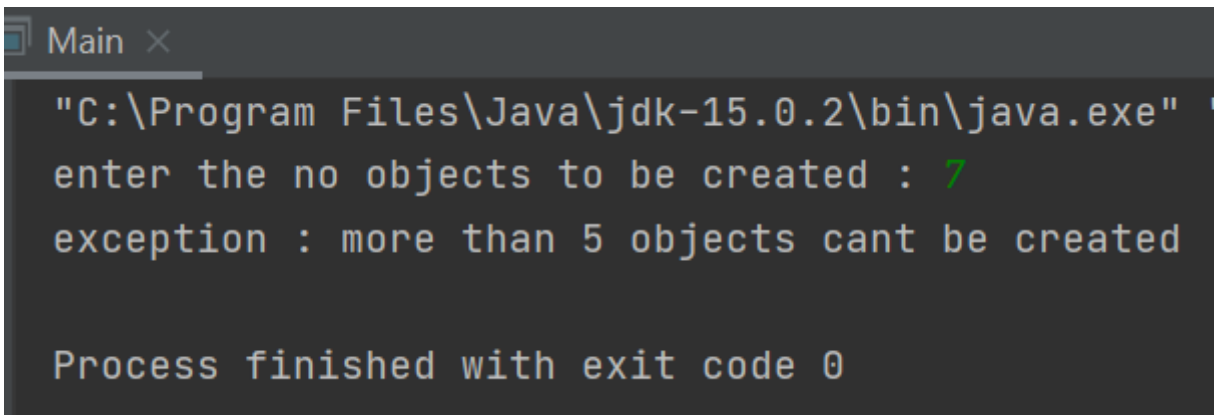
```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-ja
enter the value of k: 6
exception found the file has more than 6 characters

Process finished with exit code 0
```

without exception

```
"C:\Program Files\Java\jdk-15.0.2\b
enter the value of k: 10000
no exception found

Process finished with exit code 0
```

10. It is known that we can create a number of objects for a class. But from the requirement analysis, we come to know that only 5 objects have to be created for a particular class. So, if we try to create more than 5 objects to a class throw a custom exception manually to JVM. Print the corresponding exception message onto the output screen

```java
 package com.deepak;
import java.util.Scanner;

class myexception extends Exception{
    public String  toString(){
        return "exception : more than 5 objects cant be created";
    }
}
class dummy
{
    private void pprint()
    {
        System.out.println("This is parent class");
    }
}


public class Main {
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        int n;
        System.out.print("enter the no objects to be created : ");
        try
        {
            n=sc.nextInt();
            if (n>5)
                throw new myexception();
        }
        catch (myexception e)
        {
            System.out.println(e);
        }
    }
}
```



11. You have been assigned to create a student database of some college. So for a class named Studentinfo create several objects each object is for one student (which means you have to create an array of objects). Studentinfo class is a different class the main class is different

```java
package com.deepak;
```

```
import java.util.Scanner;
import java.io.*;

class student{
   int id;
   String name, department,college;
   student(int Id,String Name, String Department, String College)
   {
      id=Id;
      name=Name;
      department=Department;
      college=College;
   }
}

class studentinfo{
   student stu[];
   studentinfo(int x)
   {
      stu=new student[x];
   }
   void take_input()
   {
      System.out.println("enter the id, name, department, college for students");
      Scanner sc=new Scanner(System.in);
      int id;
      String name, department,college;
      for (int i=0;i<stu.length;i++)
      {
         int f=0;
         id=sc.nextInt();
         name=sc.next();
         department=sc.next();
         college=sc.next();
         for (int j=0;j<i;j++)
         {
            if (stu[j].id==id)
            {
               i--;
               f=1;
               System.out.println("id already exists . enter a different id ");
               break;
            }
         }
         if (f==0)
         {
            stu[i]=new student(id,name,department,college);
            System.out.println("value entered");
         }
         else
            continue;
      }

   }

   void print()
   {
      System.out.println();
      System.out.println("the student info in the database are :-");
      System.out.println("id\t"+ "name\t"+"department\t"+"college\t");
      for (int i=0;i<stu.length;i++)
```

```
        {
            System.out.println(stu[i].id+"\t"+stu[i].name+"\t"+stu[i].department+"\t"+stu[i].college);
        }
    }
}

public class Main {
    public static void main(String[] args) throws IOException{
        Scanner sc=new Scanner(System.in);
        int n;
        System.out.print("enter the no of students : ");
        n=sc.nextInt();
        studentinfo sol=new studentinfo(n);
        sol.take_input();
        sol.print();
    }
}
```

```
enter the no of students : 5
enter the id, name, department, college for students
1 bhuvam mech dtu
value entered
2 ashish civil mnit
value entered
2 samay biotech mnit
id already exists . enter a different id
2 samay biotech mnit
id already exists . enter a different id
3 samay biotech mnit
value entered
4 ajay chem nsit
value entered
5 magnus cse iitb
value entered

the student info in the database are :-
id   name     department   college
1    bhuvam   mech     dtu
2    ashish   civil    mnit
3    samay    biotech  mnit
4    ajay     chem     nsit
5    magnus   cse iitb
```

12. Write a java program to illustrate 4 different possible NullPointerException cases.

```
package com.deepak;
import java.util.Scanner;
import java.io.*;

public class Main {
   public static void main(String[] args) throws IOException{
      Scanner sc=new Scanner(System.in);
      String s=null;
      //Invoking a method from a null object.
      try
      {
         System.out.println(s.length());
      }
      catch (NullPointerException e)
      {
         System.out.println(e);
      }
      //Accessing or modifying a null object's field
      try
      {

         char c=s.charAt(3);
      }
      catch (NullPointerException e)
      {
         System.out.println(e);
      }
      //Accessing or modifying the slots of null object, as if it were an array.
      try{
      int x[]=null;
      x[2]=7;}
      catch (NullPointerException e)
      {
         System.out.println(e);
      }
   }
}
```

```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaagent:C:\Users\Deepak\AppData\Local
java.lang.NullPointerException: Cannot invoke "String.length()" because "s" is null
java.lang.NullPointerException: Cannot invoke "String.charAt(int)" because "s" is null
java.lang.NullPointerException: Cannot store to int array because "x" is null

Process finished with exit code 0
```