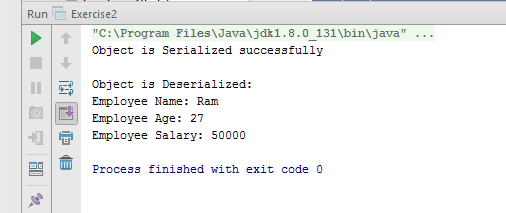
1. **Write a program to copy and paste a file to other location**

**import** java.io.\*;  
**public class** Exercise1 {  
 **public static void** main(String[] args) **throws** IOException{  
  
 BufferedReader reader = **null**;  
 BufferedWriter writer = **null**;  
 **try** {  
 FileReader fileReader = **new** FileReader(**"input.txt"**);  
 FileWriter fileWriter = **new** FileWriter(**"output.txt"**);  
 reader = **new** BufferedReader(fileReader);  
 writer = **new** BufferedWriter(fileWriter);  
 **int** c;  
 **while** ((c = reader.read()) != -1) {  
 writer.write(c);  
  
 }  
 }  
  
 **finally** {  
 **if**(reader != **null**){  
 reader.close();  
 }  
 **if**(writer != **null**){  
 writer.close();  
  
 }  
 }  
  
 }  
}

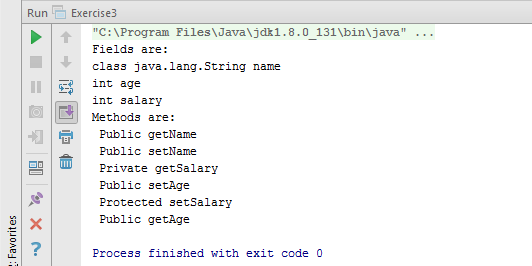
**2. Write a program to serialize a class Employee with property name and then deserialize it**

**import** java.io.\*;  
**public class** Exercise2 {  
 **public static void** main(String[] args){  
  
 Employee e = **new** Employee();  
 e.**name** = **"Ram"**;  
 e.**age** = 27;  
 e.**salary** = 50000;  
 **try** {  
  
 FileOutputStream fileOutputStream = **new** FileOutputStream(**"EmployeeObj.txt"**);  
 ObjectOutputStream outputStream = **new** ObjectOutputStream(fileOutputStream);  
 outputStream.writeObject(e);  
 fileOutputStream.close();  
 outputStream.close();  
 System.***out***.println(**"Object is Serialized successfully\n"**);  
  
 *// Deserializing object* Employee emp = **null**;  
 FileInputStream fileInputStream = **new** FileInputStream(**"EmployeeObj.txt"**);  
 ObjectInput inputStream = **new** ObjectInputStream(fileInputStream);  
 emp = (Employee)inputStream.readObject();  
 fileInputStream.close();  
 inputStream.close();  
 System.***out***.println(**"Object is Deserialized: "**);  
 System.***out***.println(**"Employee Name: "**+emp.**name**);  
 System.***out***.println(**"Employee Age: "**+emp.**age**);  
 System.***out***.println(**"Employee Salary: "**+emp.**salary**);  
 }  
 **catch** (ClassNotFoundException c){  
 c.printStackTrace();  
  
 }  
 **catch**(IOException i){  
 i.printStackTrace();  
  
 }  
  
 }  
}



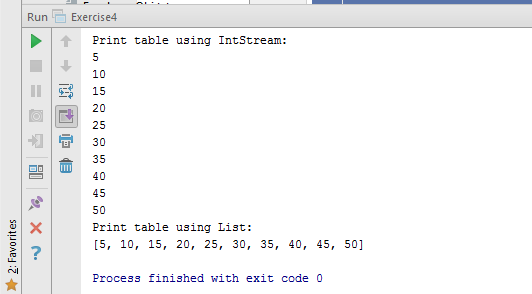
**3. Write a program to read all field and method names of a class using reflection**

**import** java.io.\*;  
**import** java.lang.reflect.Field;  
**import** java.lang.reflect.Method;  
**import** java.lang.reflect.Modifier;  
  
**public class** Exercise3 {  
 **public static void** main(String[] args) {  
  
 Employee e = **new** Employee();  
 e.**name** = **"Ram"**;  
 e.**age** = 27;  
 e.**salary** = 50000;  
  
 Field[] fields = e.getClass().getDeclaredFields();  
 System.***out***.println(**"Fields are: "**);  
 **for** (Field field:fields)  
 {  
 System.***out***.println(field.getType()+**" "**+field.getName());  
 }  
 System.***out***.println(**"Methods are: "**);  
 Method[] methods = e.getClass().getDeclaredMethods();  
 **for** (Method method :methods) {  
 **if**(Modifier.*isPublic*(method.getModifiers())){  
 System.***out***.print( **" Public "**);  
 }  
 **else if**(Modifier.*isProtected*(method.getModifiers())){  
 System.***out***.print( **" Protected "**);  
 }  
 **else if**(Modifier.*isPrivate*(method.getModifiers())){  
 System.***out***.print( **" Private "**);  
 }  
 System.***out***.println(method.getName());  
  
 }  
 }  
}



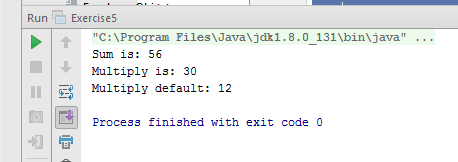
**4. Write a program to print a table of any number using lambda expression**

**import** java.util.Arrays;  
**import** java.util.List;  
**import** java.util.Scanner;  
**import** java.util.stream.Collectors;  
**import** java.util.stream.IntStream;  
  
  
**public class** Exercise4 {  
 **public static void** main(String[] args) {  
 List<Integer> list = Arrays.*asList*(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);  
 System.***out***.println(**"Enter a number: "**);  
 Scanner sc = **new** Scanner(System.***in***);  
 **int** num = sc.nextInt();  
*//Using IntStream* System.***out***.println(**"Print table using IntStream: "**);  
 IntStream.*rangeClosed*(1,10).forEach(e-> System.***out***.println(e\*num));  
*//Using List* System.***out***.println(**"Print table using List: "**);  
 List Table = list.stream()  
 .map(e->e\*num)  
 .collect(Collectors.*toList*());  
  
 System.***out***.println(Table);  
  
  
  
 }  
}



**5. Write a program having interface with default and static methods and call them also in your main method**

**interface** Exercise {  
 **public static int** sum(**int** a, **int** b){  
 System.***out***.println(**"Sum is: "**+a+b);  
 **return** a+b;  
 }  
 **public default void** mul(**int** a, **int** b){  
 System.***out***.println(**"Multiply is: "**+a\*b);  
  
 }  
}  
  
**public class** Exercise5 **implements** Exercise{  
  
 **public void** mul(**int** a, **int** b){  
 Exercise.**super**.mul(5,6);  
 System.***out***.println(**"Multiply default: "**+a\*b);  
  
 }  
 **public static void** main(String[] args) {  
  
 Exercise5 obj = **new** Exercise5();  
 Exercise.*sum*(5,6);  
 obj.mul(3,4);  
  
  
  
 }  
}



**6. Create a class Employee with name and age. Then write a program to print all those employees whose name start with n and age is greater then 24.**

**public class** Exercise6 {  
 **public static void** main(String[] args) {  
 Employee []obj = **new** Employee[5];  
 obj[0] = **new** Employee(**"Ram"**, 25,50000);  
 obj[1] = **new** Employee(**"Shyam"**, 27,55000);  
 obj[2] = **new** Employee(**"Karan"**,23,45000);  
 obj[3] = **new** Employee(**"Nishant"**,25,57000);  
 obj[4] = **new** Employee(**"Naman"**, 22,40000);  
  
 **for** (Employee e: obj) {  
 **if**(e.**name**.toLowerCase().charAt(0) == **'n'** && e.**age** > 24){  
 System.***out***.println(**"Employee Name: "**+e.**name**+**"\n"**+**"Age: "**+e.**age**+**"\n"**+**"Salary: "**+e.**salary**+**"\n"**);  
 }  
  
 }  
  
  
 }  
}

