1. Write a program to create a new file named test.txt ?

Program :

```
package Day10;
import java.io.File;
import java.io.IOException;
public class Create_file {
    public static void main(String[] args) {
        try {
            File file = new File("test.txt");
            if (file.createNewFile()) {
                System.out.println("File created: " + file.getName());
            } else {
                System.out.println("File already exists.");
            }
        } catch (IOException e) {
            System.out.println("An error occurred.");
            e.printStackTrace();
        }
    }
}
```

Ouptut : File created: test.txt

2.Write a java program to check whether a file exists at agiven path ?

Program :

```
package Day10;
import java.io.File;
public class File_exists {
    public static void main(String[] args) {
        File file = new File("test.txt");
        if (file.exists()) {
            System.out.println("File exists at: " + file.getAbsolutePath());
    } else {
      System.out.println("File does not exist.");
    }
  }
}
```

Output : File exists at: C:\Users\k devendra\eclipse-workspace\Assignments\test.txt

3. Write a Java program to write "Hello, World!" into a file using FileWriter ?

Program :

```
package File_Handling;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
public class write_file {
        public static void main(String[] args) {
                try {
                        BufferedWriter w = new BufferedWriter(new FileWriter("Sample.txt"));
                        w.write("Hello World");
                        w.close();
                        System.out.println("Write file successfully");
                 } catch (IOException e) {
      System.out.println("Error: " + e );
      e.printStackTrace();
        }
        }
}
```

Output : Write file successfully

4. Write a program to read the content of a file line by line using BufferedReader ?

Program :

```
package Day10;
```

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class Read_File {
   public static void main(String[] args) {
      try (BufferedReader br = new BufferedReader(new FileReader("test.txt"))) {
         String line;
         while ((line = br.readLine()) != null) {
            System.out.println(line);
         }
      } catch (IOException e) {
         e.printStackTrace();
      }
   }
}
```

Output : Hello World

5. Write a program to append a line of text to an existing file ?

Program :

```
package Day10;
import java.io.FileWriter;
import java.io.IOException;
public class Append_File {
   public static void main(String[] args) {
      try (FileWriter fw = new FileWriter("test.txt", true)) {
         fw.write("\nThis is an appended line.");
         System.out.println("Appended to file.");
      } catch (IOException e) {
         e.printStackTrace();
      }
```

```
    }
}
```

Output : Appended to file.

6. Write a program to count the number of lines, words, and characters in a file.

Program :

```
package Day10;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
public class Count {
    public static void main(String[] args) {
        int lines = 0, words = 0, chars = 0;
        try (BufferedReader br = new BufferedReader(new FileReader("test.txt"))) {
            String line;
            while ((line = br.readLine()) != null) {
                lines++;
                chars += line.length();
                words += line.split("\\s+").length;
            }
            System.out.println("Lines: " + lines);
            System.out.println("Words: " + words);
            System.out.println("Characters: " + chars);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output : Lines: 2

Words: 7

Characters: 36

7. Write a program to copy content from one file to another using FileReader and FileWriter.

Program :

```
package Day10;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class Copy_File {

    public static void main(String[] args) {

        try (FileReader fr = new FileReader("test.txt");

             FileWriter fw = new FileWriter("copy.txt")) {

            int c;

            while ((c = fr.read()) != -1) {

                fw.write(c);

            }

            System.out.println("File copied successfully.");

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}
```

Output : File copied successfully.

8. Write a program that lists all the files in a directory.

Program :

```
package Day10;

import java.io.File;

public class All_Files {

    public static void main(String[] args) {

        File dir = new File(".");

        String[] files = dir.list();
        if (files != null) {
            for (String file : files) {
                System.out.println(file);
            }
        }
    }
}
```

Output : .classpath

```
.project
.settings
bin
copy.txt
src
student.ser
student.txt
test.txt
userinput.txt
```

9. Write a program to filter and display only .txt files from a folder using FilenameFilter.

Program :

```
import java.io.File;

import java.io.FilenameFilter;

public class FilterTxtFiles {

    public static void main(String[] args) {

        File dir = new File(".");

        FilenameFilter filter = (f, name) -> name.endsWith(".txt");

        String[] txtFiles = dir.list(filter);

        if (txtFiles != null) {

            for (String name : txtFiles) {

                System.out.println(name);

            }

        }

    }

}
```

Output : copy.txt

student.txt
test.txt
userinput.txt

10. Write a program to serialize and deserialize a Student object to and from a file.

Program :

```
package File_Handling;

import java.io.*;

public class Student implements Serializable {

    int rollno;

    String name;

    Student(int id, String name) {

        this.rollno = id;

        this.name = name;

    }

    void display() {

        System.out.println("Rollno : " + rollno + " Name : "  +name );

    }

}

package File_Handling;

import java.io.*;

public class Se_data {

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

        Student s1 = new Student(101, "Virat");

        FileOutputStream fos = new FileOutputStream("student.ser");

        ObjectOutputStream oos = new ObjectOutputStream(fos);


        oos.writeObject(s1);

        oos.close();

        fos.close();

        System.out.println("Serialized done successfully");
    }
}
package File_Handling;
import java.io.*;
public class De_data {
    public static void main(String[] args) {
        try {
            FileInputStream fis = new FileInputStream("student.ser");
            ObjectInputStream ois = new ObjectInputStream(fis);
            Student s1 = (Student) ois.readObject();
            ois.close();
            fis.close();
            System.out.println("Deserialization Done successfully");
            s1.display();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
Output : Serialized done successfully
Deserialization Done successfully
Rollno : 101 Name : Virat
```

11. Write a program to read a file using Scanner and display the tokens.

Program :

```
package Day10;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
public class Read_File_Scanner {
    public static void main(String[] args) {
        try (Scanner sc = new Scanner(new File("test.txt"))) {
            while (sc.hasNext()) {
                System.out.println(sc.next());
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

```
Output : Hello
World
This
is
an
appended
line.
```

12. Write a program to search for a specific word in a file and count its occurrences.

Program :

```
package Day10;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
public class Search {
    public static void main(String[] args) {
        String wordToFind = "Hello";
        int count = 0;
        try (BufferedReader br = new BufferedReader(new FileReader("test.txt"))) {
            String line;
            while ((line = br.readLine()) != null) {
                String[] words = line.split("\\s+");
                for (String w : words) {
                    if (w.equalsIgnoreCase(wordToFind)) {
                        count++;
                    }
                }
            }
            System.out.println("Occurrences of '" + wordToFind + "': " + count);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output : Occurrences of 'Hello': 1

13. Write a program to create, move, and delete a file using Files and Paths.

Program :

```
package Day10;
import java.io.IOException;
import java.nio.file.*;
public class File_Ops {
    public static void main(String[] args) {
        Path source = Paths.get("tempfile.txt");
        Path target = Paths.get("movedfile.txt");

        try {

            Files.createFile(source);
            System.out.println("File created: " + source);

            Files.move(source, target, StandardCopyOption.REPLACE_EXISTING);
            System.out.println("File moved to: " + target);

            Files.delete(target);
            System.out.println("File deleted: " + target);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
Output : File created: tempfile.txt
File moved to: movedfile.txt
File deleted: movedfile.txt
```

14. Write a program to read all lines of a file using Files.readAllLines() and print them.

Program :

```
package Day10;
import java.io.IOException;
import java.nio.file.*;
import java.util.List;
public class Read_Lines {
    public static void main(String[] args) {
        try {
            List<String> lines = Files.readAllLines(Paths.get("test.txt"));
            for (String line : lines) {
                System.out.println(line);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

```
Output : Hello World
This is an appended line.
```

15. Write a program to write data into a file using Files.write() and append using StandardOpenOption.APPEND.

Program :

```
package Day10;
import java.io.IOException;
import java.nio.file.*;
import java.nio.file.StandardOpenOption;
import java.util.Arrays;
public class File_append {
    public static void main(String[] args) {
        try {
            Path path = Paths.get("niofile.txt");

            Files.write(path, Arrays.asList("First line", "Second line"));

            Files.write(path, Arrays.asList("Appended line"), StandardOpenOption.APPEND);

            System.out.println("Write and append completed.");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output : Write and append completed.

16. Write a program to walk through a directory tree and display file names using Files.walk().

Program :

```
package Day10;
import java.io.IOException;
import java.nio.file.*;
public class Dir_Files {
    public static void main(String[] args) {
        try {
            Files.walk(Paths.get("."))
```

```
                .forEach(System.out::println);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output : .

```
.\.classpath
.\.project
.\.settings
.\.settings\org.eclipse.core.resources.prefs
.\.settings\org.eclipse.jdt.core.prefs
.\bin
.\bin\Day10
.\bin\Day10\Test.class
.\bin\Day11
.\bin\Day11\Test.class
.\bin\Day4_Abstraction
.\bin\Day4_Abstraction\Account.class
.\bin\Day4_Abstraction\BankInterface.class
.\bin\Day4_Abstraction\Bank_A.class
.\bin\Day4_Abstraction\Bike.class
.\bin\Day4_Abstraction\Car.class
.\bin\Day4_Abstraction\Circle.class
.\bin\Day4_Abstraction\CurrentAccount.class
.\bin\Day4_Abstraction\Drawable.class
.\bin\Day4_Abstraction\Person.class
.\bin\Day4_Abstraction\Person_A.class
.\bin\Day4_Abstraction\Phone_A.class
.\bin\Day4_Abstraction\Professor.class
.\bin\Day4_Abstraction\Rectangle.class
.\bin\Day4_Abstraction\SavingsAccount.class
.\bin\Day4_Abstraction\Shape.class
.\bin\Day4_Abstraction\Shape_A.class
.\bin\Day4_Abstraction\SmartDevice.class
.\bin\Day4_Abstraction\SmartPhone.class
.\bin\Day4_Abstraction\SmartSpeaker.class
.\bin\Day4_Abstraction\SmartWatch.class
.\bin\Day4_Abstraction\Student.class
.\bin\Day4_Abstraction\TeachingAssistant.class
.\bin\Day4_Abstraction\Triangle.class
.\bin\Day4_Abstraction\Truck.class
```

```
.\bin\Day4_Abstraction\Vehicle.class
.\bin\Day4_Abstraction\Vehicle_A.class
.\bin\Day4_Inheritance
.\bin\Day4_Inheritance\Accountant.class
.\bin\Day4_Inheritance\After_12th.class
.\bin\Day4_Inheritance\Animal.class
.\bin\Day4_Inheritance\Baleno.class
.\bin\Day4_Inheritance\BBA.class
.\bin\Day4_Inheritance\BCA.class
.\bin\Day4_Inheritance\BDS.class
.\bin\Day4_Inheritance\Cardiac.class
.\bin\Day4_Inheritance\CS.class
.\bin\Day4_Inheritance\Doctor.class
.\bin\Day4_Inheritance\Documentation.class
.\bin\Day4_Inheritance\Dog.class
.\bin\Day4_Inheritance\Education_Heirarchy.class
.\bin\Day4_Inheritance\Endo.class
.\bin\Day4_Inheritance\Engineering.class
.\bin\Day4_Inheritance\FiveSeaterPetrolFourWheeler.class
.\bin\Day4_Inheritance\FourWheeler.class
.\bin\Day4_Inheritance\Gynac.class
.\bin\Day4_Inheritance\Hospital.class
.\bin\Day4_Inheritance\Hospital1.class
.\bin\Day4_Inheritance\Hospital_Heirarchy.class
.\bin\Day4_Inheritance\Hospital_In.class
.\bin\Day4_Inheritance\IT.class
.\bin\Day4_Inheritance\MBBS.class
.\bin\Day4_Inheritance\Mech.class
.\bin\Day4_Inheritance\Medical.class
.\bin\Day4_Inheritance\Nurse.class
.\bin\Day4_Inheritance\OPD.class
.\bin\Day4_Inheritance\Operation.class
.\bin\Day4_Inheritance\Other_Courses.class
.\bin\Day4_Inheritance\Patient.class
.\bin\Day4_Inheritance\Payments.class
.\bin\Day4_Inheritance\PetrolFourWheeler.class
.\bin\Day4_Inheritance\Super_key.class
.\bin\Day4_Inheritance\Vehicle.class
.\bin\Day4_Inheritance\Vehicle_In.class
.\bin\Day4_Polymorphism
.\bin\Day4_Polymorphism\Bank.class
.\bin\Day4_Polymorphism\Bank_P.class
.\bin\Day4_Polymorphism\Bike.class
.\bin\Day4_Polymorphism\Calculator.class
.\bin\Day4_Polymorphism\Calculator_P.class
```

```
.\bin\Day4_Polymorphism\Circle.class
.\bin\Day4_Polymorphism\HDFC.class
.\bin\Day4_Polymorphism\ICICI.class
.\bin\Day4_Polymorphism\Rectangle.class
.\bin\Day4_Polymorphism\SBI.class
.\bin\Day4_Polymorphism\Shape.class
.\bin\Day4_Polymorphism\Shape_P.class
.\bin\Day4_Polymorphism\Vehicle.class
.\bin\Day4_Polymorphism\Vehicle_P.class
.\bin\Day5_Encapsulation
.\bin\Day5_Encapsulation\Rectangle.class
.\bin\Day5_Encapsulation\Student.class
.\bin\Day5_Encapsulation\Test.class
.\bin\Day5_Interface
.\bin\Day5_LambdaExpressions
.\bin\Day5_LambdaExpressions\LambdaEvenOdd.class
.\bin\Day5_LambdaExpressions\LambdaFactorial.class
.\bin\Day5_LambdaExpressions\LambdaSum.class
.\bin\Day5_LambdaExpressions\String_Empty.class
.\bin\Day5_LambdaExpressions\SumCalculator.class
.\bin\Day6
.\bin\Day6\Test.class
.\bin\Day7
.\bin\Day7\Test.class
.\bin\Day8
.\bin\Day8\ArrayListClear_Q7.class
.\bin\Day8\ArrayListCopy_Q10.class
.\bin\Day8\ArrayListIterator_Q8.class
.\bin\Day8\ArrayListRemoveElement_Q3.class
.\bin\Day8\ArrayListReverse_Q5.class
.\bin\Day8\ArrayListSearch_Q2.class
.\bin\Day8\ArrayListSort_Q4.class
.\bin\Day8\ArrayListStudent_Q9.class
.\bin\Day8\ArrayListUpdate_Q6.class
.\bin\Day8\Deque_Q1.class
.\bin\Day8\Deque_Q2.class
.\bin\Day8\Deque_Q3.class
.\bin\Day8\HashSetCities_Q1.class
.\bin\Day8\HashSetMax_Q3.class
.\bin\Day8\HashSetOps_Q2.class
.\bin\Day8\LinkedHashSet_Q1.class
.\bin\Day8\LinkedHashSet_Q2.class
.\bin\Day8\LinkedHashSet_Q3.class
.\bin\Day8\LinkedListAddEnds_Q2.class
.\bin\Day8\LinkedListBook_Q9.class
```

```
.\bin\Day8\LinkedListClone_Q10.class
.\bin\Day8\LinkedListDisplay_Q1.class
.\bin\Day8\LinkedListInsert_Q3.class
.\bin\Day8\LinkedListIterator_Q6.class
.\bin\Day8\LinkedListRemove_Q4.class
.\bin\Day8\LinkedListSearch_Q5.class
.\bin\Day8\LinkedListSort_Q7.class
.\bin\Day8\LinkedListToArrayList_Q8.class
.\bin\Day8\PriorityQueue_Q1.class
.\bin\Day8\PriorityQueue_Q2.class
.\bin\Day8\PriorityQueue_Q3.class
.\bin\Day8\Queue_Q1.class
.\bin\Day8\Queue_Q2.class
.\bin\Day8\Queue_Q3.class
.\bin\Day8\Stack_Q1.class
.\bin\Day8\Stack_Q2.class
.\bin\Day8\Stack_Q3.class
.\bin\Day8\Stack_Q4.class
.\bin\Day8\Student.class
.\bin\Day8\TreeSet_Q1.class
.\bin\Day8\TreeSet_Q2.class
.\bin\Day8\TreeSet_Q3.class
.\bin\Day8\VectorCompare_Q3.class
.\bin\Day8\VectorInteger_Q1.class
.\bin\Day8\VectorString_Q2.class
.\bin\Day8\VectorSum_Q4.class
.\bin\Day9
.\bin\Day9\AppendToFile.class
.\bin\Day9\Book.class
.\bin\Day9\Book1.class
.\bin\Day9\City.class
.\bin\Day9\CopyFile.class
.\bin\Day9\CountWords.class
.\bin\Day9\CustomObject.class
.\bin\Day9\DeleteFile.class
.\bin\Day9\DeserializeStudent.class
.\bin\Day9\Employee.class
.\bin\Day9\Employee1.class
.\bin\Day9\Employee2.class
.\bin\Day9\FileProperties.class
.\bin\Day9\Person.class
.\bin\Day9\PrintFiles.class
.\bin\Day9\Product.class
.\bin\Day9\Product1.class
.\bin\Day9\Product2.class
```

```
.\bin\Day9\ReadFile.class
.\bin\Day9\ReplaceFile.class
.\bin\Day9\ReverseFile.class
.\bin\Day9\SearchWord.class
.\bin\Day9\SerializeStudent.class
.\bin\Day9\String_Sort$1.class
.\bin\Day9\String_Sort.class
.\bin\Day9\Student.class
.\bin\Day9\Student1.class
.\bin\Day9\Student4.class
.\bin\Day9\Student_Desc.class
.\bin\Day9\UserInputFile.class
.\bin\Day9\WriteToFile.class
.\copy.txt
.\niofile.txt
.\src
.\src\Day10
.\src\Day10\Test.java
.\src\Day11
.\src\Day11\Test.java
.\src\Day4_Abstraction
.\src\Day4_Abstraction\Bank_A.java
.\src\Day4_Abstraction\Person_A.java
.\src\Day4_Abstraction\Phone_A.java
.\src\Day4_Abstraction\Shape_A.java
.\src\Day4_Abstraction\Vehicle_A.java
.\src\Day4_Inheritance
.\src\Day4_Inheritance\Education_Heirarchy.java
.\src\Day4_Inheritance\Hospital_Heirarchy.java
.\src\Day4_Inheritance\Hospital_In.java
.\src\Day4_Inheritance\Super_key.java
.\src\Day4_Inheritance\Vehicle_In.java
.\src\Day4_Polymorphism
.\src\Day4_Polymorphism\Bank_P.java
.\src\Day4_Polymorphism\Calculator_P.java
.\src\Day4_Polymorphism\Shape_P.java
.\src\Day4_Polymorphism\Vehicle_P.java
.\src\Day5_Encapsulation
.\src\Day5_Encapsulation\Rectangle.java
.\src\Day5_Encapsulation\Student.java
.\src\Day5_Encapsulation\Test.java
.\src\Day5_Interface
.\src\Day5_LambdaExpressions
.\src\Day5_LambdaExpressions\LambdaEvenOdd.java
.\src\Day5_LambdaExpressions\LambdaFactorial.java
```

```
.\src\Day5_LambdaExpressions\LambdaSum.java
.\src\Day5_LambdaExpressions\String_Empty.java
.\src\Day6
.\src\Day6\Test.java
.\src\Day7
.\src\Day7\Test.java
.\src\Day8
.\src\Day8\ArrayListClear_Q7.java
.\src\Day8\ArrayListCopy_Q10.java
.\src\Day8\ArrayListIterator_Q8.java
.\src\Day8\ArrayListRemoveElement_Q3.java
.\src\Day8\ArrayListReverse_Q5.java
.\src\Day8\ArrayListSearch_Q2.java
.\src\Day8\ArrayListSort_Q4.java
.\src\Day8\ArrayListStudent_Q9.java
.\src\Day8\ArrayListUpdate_Q6.java
.\src\Day8\Deque_Q1.java
.\src\Day8\Deque_Q2.java
.\src\Day8\Deque_Q3.java
.\src\Day8\HashSetCities_Q1.java
.\src\Day8\HashSetMax_Q3.java
.\src\Day8\HashSetOps_Q2.java
.\src\Day8\LinkedHashSet_Q1.java
.\src\Day8\LinkedHashSet_Q2.java
.\src\Day8\LinkedHashSet_Q3.java
.\src\Day8\LinkedListAddEnds_Q2.java
.\src\Day8\LinkedListBook_Q9.java
.\src\Day8\LinkedListClone_Q10.java
.\src\Day8\LinkedListDisplay_Q1.java
.\src\Day8\LinkedListInsert_Q3.java
.\src\Day8\LinkedListIterator_Q6.java
.\src\Day8\LinkedListRemove_Q4.java
.\src\Day8\LinkedListSearch_Q5.java
.\src\Day8\LinkedListSort_Q7.java
.\src\Day8\LinkedListToArrayList_Q8.java
.\src\Day8\PriorityQueue_Q1.java
.\src\Day8\PriorityQueue_Q2.java
.\src\Day8\PriorityQueue_Q3.java
.\src\Day8\Queue_Q1.java
.\src\Day8\Queue_Q2.java
.\src\Day8\Queue_Q3.java
.\src\Day8\Stack_Q1.java
.\src\Day8\Stack_Q2.java
.\src\Day8\Stack_Q3.java
.\src\Day8\Stack_Q4.java
```

```
.\src\Day8\TreeSet_Q1.java
.\src\Day8\TreeSet_Q2.java
.\src\Day8\TreeSet_Q3.java
.\src\Day8\VectorCompare_Q3.java
.\src\Day8\VectorInteger_Q1.java
.\src\Day8\VectorString_Q2.java
.\src\Day8\VectorSum_Q4.java
.\src\Day9
.\src\Day9\AppendToFile.java
.\src\Day9\Book.java
.\src\Day9\Book1.java
.\src\Day9\City.java
.\src\Day9\CopyFile.java
.\src\Day9\CountWords.java
.\src\Day9\CustomObject.java
.\src\Day9\DeleteFile.java
.\src\Day9\DeserializeStudent.java
.\src\Day9\Employee.java
.\src\Day9\Employee1.java
.\src\Day9\Employee2.java
.\src\Day9\FileProperties.java
.\src\Day9\Person.java
.\src\Day9\PrintFiles.java
.\src\Day9\Product.java
.\src\Day9\Product1.java
.\src\Day9\Product2.java
.\src\Day9\ReadFile.java
.\src\Day9\ReplaceFile.java
.\src\Day9\ReverseFile.java
.\src\Day9\SearchWord.java
.\src\Day9\SerializeStudent.java
.\src\Day9\String_Sort.java
.\src\Day9\Student.java
.\src\Day9\Student1.java
.\src\Day9\Student_Desc.java
.\src\Day9\UserInputFile.java
.\src\Day9\WriteToFile.java
.\student.ser
.\student.txt
.\test.txt
.\userinput.txt
```

17. Write a program to copy a file using Files.copy() with REPLACE_EXISTING option.

Program :

```
package Day10;
import java.io.IOException;
import java.nio.file.*;
public class Copy_File {
   public static void main(String[] args) {
      Path source = Paths.get("test.txt");
      Path target = Paths.get("test1.txt");

      try {
         Files.copy(source, target, StandardCopyOption.REPLACE_EXISTING);
         System.out.println("File copied successfully.");
      } catch (IOException e) {
         e.printStackTrace();
      }
   }
}
```

Output : File copied successfully.

18. Write a program to check and print the size of a file in bytes using Files.size().

Program :

```
package Day10;
import java.io.IOException;
import java.nio.file.*;
public class File_Size {
   public static void main(String[] args) {
      try {
         long size = Files.size(Paths.get("test.txt"));
         System.out.println("File size: " + size + " bytes");
      } catch (IOException e) {
         e.printStackTrace();
      }
```

```
    }
}
```

Output : File size: 37 bytes

19. Write a program to serialize a class Employee and store it in employee.ser.

Program :

```
package Day10;
import java.io.Serializable;
public class Employee implements Serializable{
        int id;
        String name;
        Employee(int id, String name) {
                this.id = id;
                this.name = name;
                this.toString();
        }
        public String toString() {
                return id + "   " + name;
        }
}
package Day10;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
public class Serial_data {
    public static void main(String[] args) {
        Employee emp1 = createEmployee(101, "Virat");
        Employee emp2 = createEmployee(102, "Rohit");

        try {
            FileWriter fw = new FileWriter("employee.ser");
            BufferedWriter bw = new BufferedWriter(fw);

            bw.write(emp1.toString());
            bw.newLine();
            bw.write(emp2.toString());
            bw.newLine();

            bw.close();
            fw.close();
            System.out.println("Done");
        } catch (IOException e) {
            System.out.println("Failed");
            System.out.println(e);
        }
    }

    static Employee createEmployee(int id, String name) {
        return new Employee(id, name);
    }
}
```

Output : Done

20. Write a program to deserialize the employee.ser file and display the object data

Program :

```
package Day10
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
public class Deserial_data {
    public static void main(String[] args) {
        try {
            FileReader fr = new FileReader("employee.ser");
            BufferedReader br = new BufferedReader(fr);
            String line;
            System.out.println("Employee Data from file:");
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
            br.close();
            fr.close();
        } catch (IOException e) {
            System.out.println("Failed to read file");
            System.out.println(e);
        }
    }
}
```

```
Output : Employee Data from file:
101   Virat
102   Rohit
```