8) Polymorphism refers to a ability of a class to provide different implementations of a method, one example used in this lab exercise is when method BabySound() is called on Baby object, it will return the Baby Sound is Neutral, when the same method is called on a Patient object, it will return the Baby Sound is Sick instead.

| **Test ID** | **Test description/justification – what is the test for and why this particular test.** | **Actual data for this test** | **Expected output** | **Actual desk check result when desk check is carried out** | **Desk check outcome – Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 1 | Test if baby identical | Baby Test Age 1  Baby Test Age 1 | Baby is identical | Baby is identical | Pass |
| 2 | Test if program prints out patient ID | Patient ID 29 | Id: 29 | Id: 29 | Pass |
| 3 | Test if BabySound is printed | PatientObj.BabySound() | Baby Sound is Sick | Baby Sound is Sick | Pass |
| 4 | Test if information is output to baby.txt | try {  outputStream = new PrintWriter(new File("baby.txt"));  } catch (FileNotFoundException e) {  System.out.println("Can't open file baby.txt");  }  System.out.println(obj.displayBaby(arr));  outputStream.write(obj.displayBaby(arr)); | Name: Test Age: 1 Id: 29  Baby Sound is Sick  Name: Test Age: 1  Baby Sound is Neutral  Name: Peter Age: 2  Baby Sound is Happy  Name: Dave Age: 3  Baby Sound is Neutral | Name: Test Age: 1 Id: 29  Baby Sound is Sick  Name: Test Age: 1  Baby Sound is Neutral  Name: Peter Age: 2  Baby Sound is Happy  Name: Dave Age: 3  Baby Sound is Neutral | Pass |