Q 1 Write a Java class Car with the following attributes: make, model, year, and color. Implement a constructor that initializes these attributes when an object of the Car class is created. Write a main method to create an instance of Car and display its attributes.  
Enhance the Car class from the previous question by adding a parameterized constructor that takes values for make, model, year, and color as arguments. Demonstrate the use of this constructor in the main method  
 **package** java\_FinalAssignment.Que1;

SANJANA KUMBHAR  
FINAL ASSIGNMENT OF JAVACORE  
RFID-242023028

**public** **class** Car {

String make;

String model;

**int** year;

String color;

Car(){

make="unknown";

model="base";

year=0;

color="unknown";

}

Car(String make, String model,**int** year,String color){

**this**.make=make;

**this**.model=model;

**this**.year=year;

**this**.color=color;

}

**public** **void** display() {

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*car details\*\*\*\*\*\*\*\*\*\*\*\*");

System.***out***.println("make:-"+make);

System.***out***.println("model:-"+model);

System.***out***.println("year:-"+year);

System.***out***.println("color:- "+color);

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Car breeza = **new** Car("maruti suzuki","breeza",2022,"silver");

breeza.display();

Car nexon = **new** Car("tata","nexon",2023,"white");

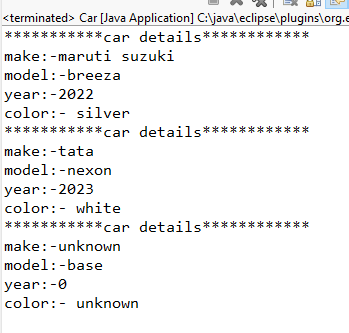
nexon.display();

Car test = **new** Car();

test.display();

}

}

  
Q 2 Create a class named Book with the following attributes:  
è title (String): to store the title of the book.  
è author (String): to store the author of the book.  
è isbn (String): to store the ISBN number of the book.  
Implement a constructor to initialize these attributes.  
·         In the main method, create an array named library that can hold up to 5 Book objects.  
·         Initialize the library array with different Book objects. You can choose any books you like or create fictional ones.  
·         Write a method named displayLibrary that takes the library array as a parameter and displays the details of each book in the array.  
·         Write a method named searchBook that takes the library array and a book title as parameters. This method should search for the book with the given title in the library array and display its details if found. If the book is not found, it should print a message indicating that the book is not in the library.

**Book.java**

**package** java\_FinalAssignment.Que2;

**public** **class** Book {

String title;

String author;

String isbn;//): to store the ISBN number of the book.

Book(){

title="Unknown";

author="Unknown";

isbn="000000";

}

Book(String title,String author,String isbn){

**this**.title=title;

**this**.author=author;

**this**.isbn=isbn;

}

**public** **void** displayDetails() {

System.***out***.println("Title: " + title);

System.***out***.println("Author: " + author);

System.***out***.println("ISBN: " + isbn);

}

}

**Library.java**

**package** java\_FinalAssignment.Que2;

**public** **class** Library{

**public** **static** **void** displayLibrary (Book[] library) {

**for**(Book b:library) {

b.displayDetails();

System.***out***.println();

}

}

**public** **static** **void** searchBook(Book[] library, String title) {

**boolean** found = **false**;

**for**(Book b:library) {

**if** (b.title.equalsIgnoreCase(title)) {

b.displayDetails();

found = **true**;

**break**; // Exit the loop once the book is found

}

}

**if**(!found) {

System.***out***.println("Book not found! ");

}

}

**public** **static** **void** main(String[] args) {

Book[] library= **new** Book[5];

library[0] = **new** Book("book1", "author1", "9780061120084");

library[1] = **new** Book("book2", "author2", "9780451524935");

library[2] = **new** Book("book3", "author3", "9780743273565");

library[3] = **new** Book("book4", "author4", "9781503280786");

library[4] = **new** Book("book5", "author5", "9780316769488");

System.***out***.println("Library Inventory:");

*displayLibrary*(library);

System.***out***.println("\nSearching for 'book5':");

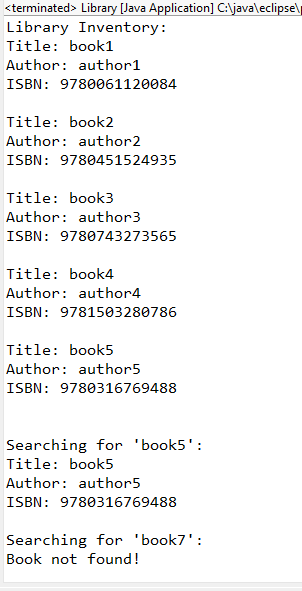
*searchBook*(library, "book5");

System.***out***.println("\nSearching for 'book7':");

*searchBook*(library, "book7");

}

}

  
  
   
Q 3 Create a class named Student with the following attributes:  
1)      name (String): to store the name of the student.  
2)      id (int): to store the student ID.  
3)      grade (double): to store the grade of the student.  
Implement a constructor to initialize these attributes.  
   
è In the main method, create an array named studentArray that can hold up to 10 Student objects.  
è Initialize the studentArray array with different Student objects. You can use fictional student data for this.  
è Write a method named displayStudents that takes the studentArray array as a parameter and displays the details of each student in the array.  
  
è Write a method named sortStudents that takes the studentArray array as a parameter and sorts the array based on the grades of the students in ascending order. You can use any sorting algorithm of your choice (e.g., bubble sort )  
**package** java\_FinalAssignment.que3;

**public** **class** Student {

String name;

**int** id;

**double** grade;

Student(){

name="Unknown";

id=0;

grade=0.00;

}

Student(String name,**int** id,**double** grade){

**this**.name=name;

**this**.id=id;

**this**.grade=grade;

}

**public** **void** displayDetails() {

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

System.***out***.println("ID: " + id);

System.***out***.println("Grade: " + grade);

System.***out***.println();

}

**public** **static** **void** displayStudents(Student[]studentArray) {

**for**(Student s:studentArray) {

s.displayDetails();

}

}

**public** **static** **void** sortStudents (Student[]studentArray ) {

// Bubble Sort algorithm

**for**(**int** i=0;i<studentArray.length-1;i++) {

**for**(**int** j=0;j<studentArray.length-1-i;j++) {

**if** (studentArray[j].grade > studentArray[j + 1].grade) {

// Swap the students

Student temp = studentArray[j];

studentArray[j] = studentArray[j + 1];

studentArray[j + 1] = temp;

}

}

}

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Student[]studentArray = **new** Student[10];

studentArray[0]= **new** Student("Sanjana",1,97.23);

studentArray[1]= **new** Student("Rishabh",2,94.29);

studentArray[2]= **new** Student("Akarsh",3,92.55);

studentArray[3]= **new** Student("Suman",4,67.56);

studentArray[4]= **new** Student("Ashwini",5,87.98);

studentArray[5]= **new** Student("Shashank",6,65.55);

studentArray[6]= **new** Student("Gaurang",7,79.78);

studentArray[7]= **new** Student("Pawan",8,91.67);

studentArray[8]= **new** Student("Kishor",9,78.53);

studentArray[9]= **new** Student("Sneha",10,79.93);

System.***out***.println("Before Sorting (by Grade):");

*displayStudents*(studentArray);

// Sort the students by grades in ascending order

*sortStudents*(studentArray);

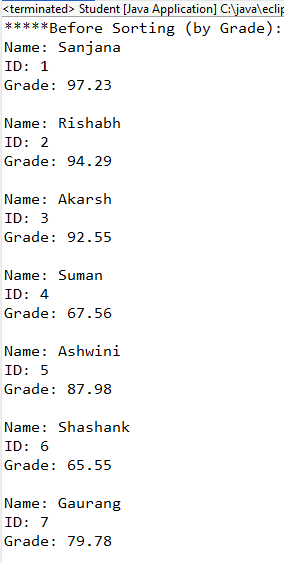
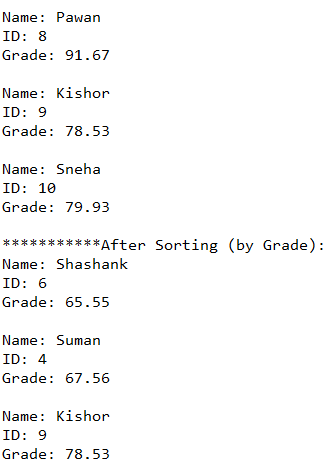
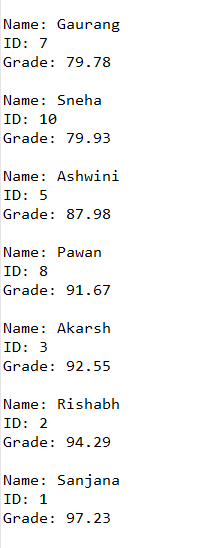
// Display the students after sorting

System.***out***.println("After Sorting (by Grade):");

*displayStudents*(studentArray);

}

}

  0  
  
Q 4  Write below mention question on paper and take picture of it and paste snapshot in above word file   
  
a) which version of java you used \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
b) Types of variables in java \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
c) Differentiate method overloading and overriding   
d) Explain constructor and types of constructor in java   
e) what is constructor chaining ?  
f) Explain concept of priority Queue .  
g) Explain multithreading and  synchronization in java .  
i) Explain dead lock situation   
j) explain recursion using any one example  
k) Explain JDK ,JRE and JVM   
L) Explain Object oriented programming   
M) explain use of lambda expression

**REFER ANOTHER PDF OF SCAN IMAGES OF BOOK PAGES**