**Question 01:**

Create a Java program defining a **Person** class with the following specifications:

* Two private fields: **name** (String) and **age** (int).
* Two constructors: a default constructor and a parameterized constructor that takes a name and an age.
* Getter and setter methods for both fields.
* An overridden **toString()** method that returns a string representation of the person's name and age.
* In the **main** method, create an instance of the **Person** class with the name "Alice" and age 28. Print the information using the **toString()** method.
* Use the setter methods to update the person's name to "Bob" and age to 35. Print the updated information.

Write the Java program for the above specifications and include the output that would be generated when the program is executed.

**Sample Answer:**

public class Person {

// Fields

private String name;

private int age;

// Constructors

public Person() {

// Default constructor

}

public Person(String name, int age) {

this.name = name;

this.age = age;

}

// Getters and Setters

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

// toString() method override

@Override

public String toString() {

return "Person [name=" + name + ", age=" + age + "]";

}

}

public class PersonTest {

public static void main(String[] args) {

// Using the default constructor

Person person1 = new Person();

person1.setName("Alice");

person1.setAge(28);

System.out.println("Person 1: " + person1.toString());

// Using the parameterized constructor

Person person2 = new Person("Bob", 35);

System.out.println("Person 2: " + person2.toString());

}

}

**Question 02:**

Create a Java program for a **Book** class with the following specifications:

1. Two private fields: **title** (String) and **author** (String).
2. Two constructors: a default constructor and a parameterized constructor that takes a title and an author.
3. Getter and setter methods for both fields.
4. An overridden **toString()** method that returns a string representation of the book's title and author.
5. In the **main** method, create an array of **Book** objects to store information about three books.
6. Initialize the array with different books using the parameterized constructor.
7. Use a loop to print the information of each book using the **toString()** method.

Write the Java program for the above specifications and include the output that would be generated when the program is executed.

**Sample Answer:**

// Book class

public class Book {

// Fields

private String title;

private String author;

// Constructors

public Book() {

// Default constructor

}

public Book(String title, String author) {

this.title = title;

this.author = author;

}

// Getters and Setters

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

// toString() method override

@Override

public String toString() {

return "Book [title=" + title + ", author=" + author + "]";

}

}

// BookTest class

public class BookTest {

public static void main(String[] args) {

// Creating individual Book objects

Book book1 = new Book("The Great Gatsby", "F. Scott Fitzgerald");

Book book2 = new Book("To Kill a Mockingbird", "Harper Lee");

Book book3 = new Book("1984", "George Orwell");

// Printing information of each book

System.out.println(book1.toString());

System.out.println(book2.toString());

System.out.println(book3.toString());

}

}

**Question 03:**

Create a Java program for a **Car** class with the following specifications:

1. Two private fields: **brand** (String) and **year** (int).
2. Two constructors: a default constructor and a parameterized constructor that takes a brand and a year.
3. Getter and setter methods for both fields.
4. An overridden **toString()** method that returns a string representation of the car's brand and year.
5. In a separate class (**CarTest**), in the **main** method:
   * Create an instance of the **Car** class with the brand "Toyota" and year 2022. Print the information using the **toString()** method.
   * Use the setter methods to update the car's brand to "Honda" and year to 2023. Print the updated information.

Write the Java program for the above specifications and include the output that would be generated when the program is executed.

**Sample Answer:**

// Car class

public class Car {

// Fields

private String brand;

private int year;

// Constructors

public Car() {

// Default constructor

}

public Car(String brand, int year) {

this.brand = brand;

this.year = year;

}

// Getters and Setters

public String getBrand() {

return brand;

}

public void setBrand(String brand) {

this.brand = brand;

}

public int getYear() {

return year;

}

public void setYear(int year) {

this.year = year;

}

// toString() method override

@Override

public String toString() {

return "Car [brand=" + brand + ", year=" + year + "]";

}

}

// CarTest class

public class CarTest {

public static void main(String[] args) {

// Creating an instance of the Car class

Car car1 = new Car("Toyota", 2022);

System.out.println("Original information: " + car1.toString());

// Using setters to update information

car1.setBrand("Honda");

car1.setYear(2023);

System.out.println("Updated information: " + car1.toString());

}

}

Output:

Original information: Car [brand=Toyota, year=2022]

Updated information: Car [brand=Honda, year=2023]

**Question 04:**

Create a Java program for a **Movie** class with the following specifications:

1. Five private fields: **title** (String) , **genre** (String), String **title**, String **director**, int **releaseYear**, String **genre**, double **rating**, int **duration**, String **language**, boolean **isReleased** .
2. Two constructors: a default constructor and a parameterized constructor that takes all the fields.
3. Getter and setter methods for both fields.
4. An overridden **toString()** method that returns a string representation of the movie's title, genre, language, duration and rating.
5. In a separate class (**MovieTest**), in the **main** method:
   * Create an instance of the **Movie** class with the title "Inception" and genre "Sci-Fi". Print the information using the **toString()** method.
   * Use the setter methods to update the movie's title to "The Matrix" and genre to "Action". Print the updated information.

Write the Java program for the above specifications and include the output that would be generated when the program is executed.

**Sample Answer:**

// Movie class

public class Movie {

// Fields

private String title;

private String director;

private int releaseYear;

private String genre;

private double rating;

private int duration; // in minutes

private String language;

private boolean isReleased;

// Constructors

public Movie() {

// Default constructor

}

public Movie(String title, String director, int releaseYear, String genre, double rating,

int duration, String language, boolean isReleased) {

this.title = title;

this.director = director;

this.releaseYear = releaseYear;

this.genre = genre;

this.rating = rating;

this.duration = duration;

this.language = language;

this.isReleased = isReleased;

}

// Getters and Setters

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getDirector() {

return director;

}

public void setDirector(String director) {

this.director = director;

}

public int getReleaseYear() {

return releaseYear;

}

public void setReleaseYear(int releaseYear) {

this.releaseYear = releaseYear;

}

public String getGenre() {

return genre;

}

public void setGenre(String genre) {

this.genre = genre;

}

public double getRating() {

return rating;

}

public void setRating(double rating) {

this.rating = rating;

}

public int getDuration() {

return duration;

}

public void setDuration(int duration) {

this.duration = duration;

}

public String getLanguage() {

return language;

}

public void setLanguage(String language) {

this.language = language;

}

public boolean isReleased() {

return isReleased;

}

public void setReleased(boolean released) {

isReleased = released;

}

// toString() method override

@Override

public String toString() {

return "Movie{" +

"title='" + title + '\'' +

", director='" + director + '\'' +

", releaseYear=" + releaseYear +

", genre='" + genre + '\'' +

", rating=" + rating +

", duration=" + duration +

", language='" + language + '\'' +

", isReleased=" + isReleased +

'}';

}

}

// MovieTest class

public class MovieTest {

public static void main(String[] args) {

// Creating an instance of the Movie class

Movie movie1 = new Movie("Inception", "Christopher Nolan", 2010, "Sci-Fi", 8.8, 148, "English", true);

System.out.println("Original information:\n" + movie1.toString());

// Using setters to update information

movie1.setTitle("The Dark Knight");

movie1.setDirector("Christopher Nolan");

movie1.setReleaseYear(2008);

movie1.setGenre("Action");

movie1.setRating(9.0);

movie1.setDuration(152);

movie1.setLanguage("English");

movie1.setReleased(true);

System.out.println("\nUpdated information:\n" + movie1.toString());

}

}