1. Source code

Aims.java:

```
public class Aims {

	public static void main(String[] args) {
		//Create a new cart
		Cart anOrder = new Cart();

		//Create new DVD objects and add them to the cart
		DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King",
				"Animation", "Roger Allers", 87, 19.95f);
		anOrder.addDigitalVideoDisc(dvd1);

		DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star Wars",
				"Science Fiction", "George Lucas", 87, 24.95f);
		anOrder.addDigitalVideoDisc(dvd2);

		DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin",
				"Animation", 18.99f);
		anOrder.addDigitalVideoDisc(dvd3);

		//To check if the remove method runs properly
		anOrder.removeDigitalVideoDisc(dvd2);
		anOrder.removeDigitalVideoDisc(dvd2);

		//Print total cost of the items in the cart
		System.out.println("Total cost is: ");
		System.out.println(anOrder.totalCost());
	}

}
```

Cart.java:

```
public class Cart {
	private int qtyOrdered = 0;
	public static final int MAX_NUMBER_ORDERED = 20;

	private DigitalVideoDisc itemsOrdered[] =
			new DigitalVideoDisc[MAX_NUMBER_ORDERED];

	//Add disc
	public void addDigitalVideoDisc(DigitalVideoDisc disc) {
	  if (qtyOrdered >= MAX_NUMBER_ORDERED) {
		System.out.println("The cart is full. Cannot add another
item.");
		return;
	  }
	  itemsOrdered[qtyOrdered] = disc;
	  qtyOrdered++;
	  System.out.println("The disc has been added.");
    }

	//Remove disc
	public void removeDigitalVideoDisc(DigitalVideoDisc disc) {
	    boolean found = false;
	    for (int i = 0; i < qtyOrdered; i++) {
	        if (itemsOrdered[i].equals(disc)) {
```

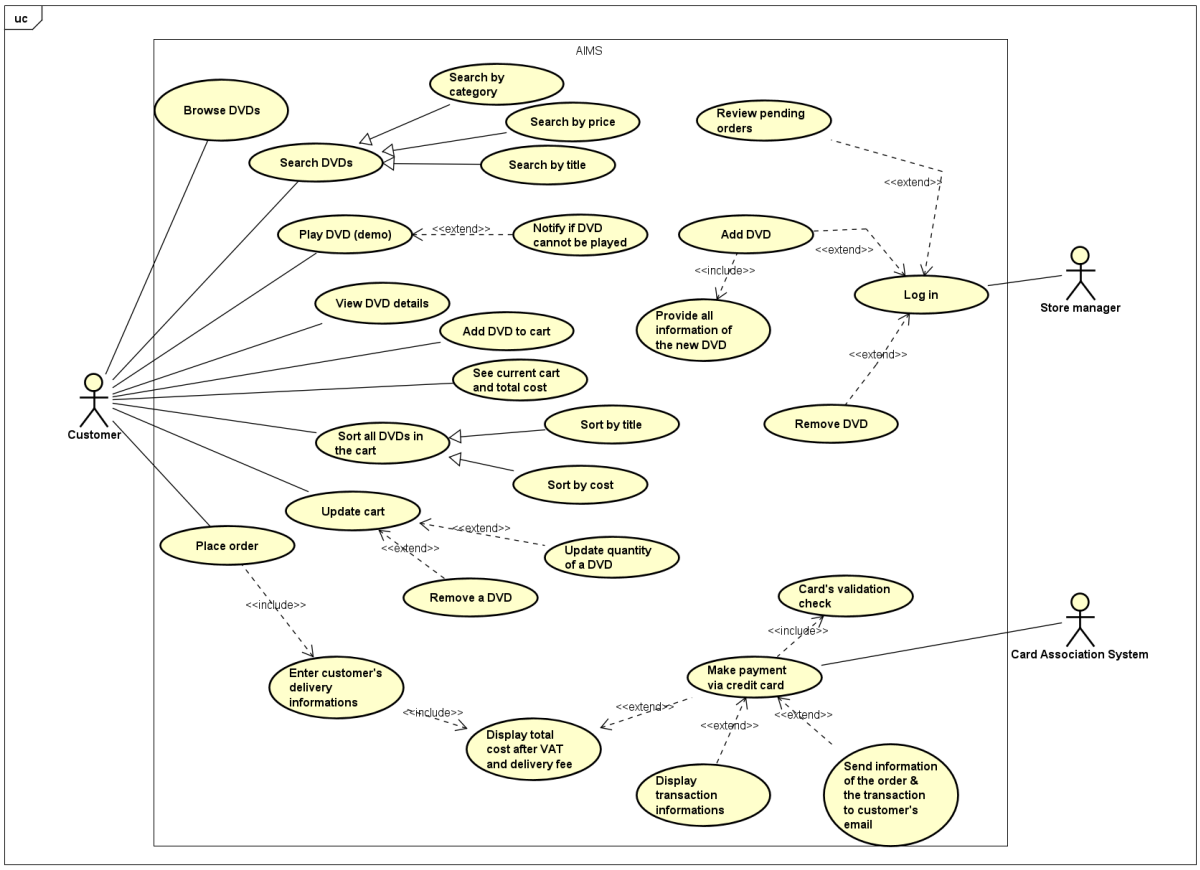
```
                found = true;
                itemsOrdered[i] = null;
                for (int j = i; j < qtyOrdered - 1; j++) {
                    itemsOrdered[j] = itemsOrdered[j + 1];
                }
                qtyOrdered--;
                System.out.println("The disc has been removed.");
                break;
            }
        }
        if (!found) {
            System.out.println("The disc was not found in the cart.");
        }
    }

    //Get total cost
    public float totalCost() {
      float total = 0.0f;

      for (int i = 0; i < qtyOrdered; i++) {
          total += itemsOrdered[i].getCost();
      }
      return total;
    }
}
```

DigitalVideoDisc.java:

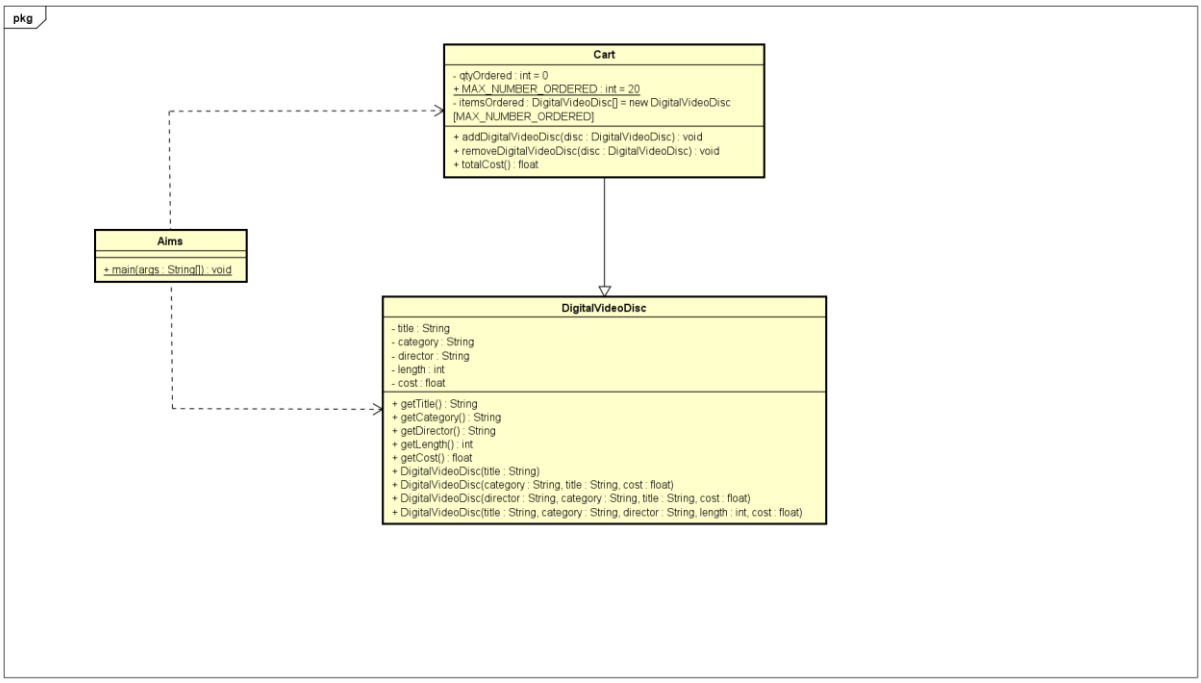
```
public class DigitalVideoDisc {
      private String title;
      private String category;
      private String director;
      private int length;
      private float cost;
      public String getTitle() {
            return title;
      }
      public String getCategory() {
            return category;
      }
      public String getDirector() {
            return director;
      }
      public int getLength() {
            return length;
      }
      public float getCost() {
            return cost;
      }

      //Create a DVD object by title
      public DigitalVideoDisc (String title) {
            super();
            this.title = title;
      }

      //Create a DVD object by category, title and cost
      public DigitalVideoDisc (String category, String title, float cost) {
            super();
            this.category = category;
            this.title = title;
```

```
            this.cost = cost;
      }


      //Create a DVD object by director, category, title and cost
      public DigitalVideoDisc (String director, String category, String
title, float cost) {
            super();
            this.director = director;
            this.category = category;
            this.title = title;
            this.cost = cost;
      }

      //Create a DVD by all attributes: title, category, director, length
and cost
      public DigitalVideoDisc (String title, String category, String
director, int length, float cost){
            super();
            this.title = title;
        this.category = category;
        this.director = director;
        this.length = length;
        this.cost = cost;
      }

}
```

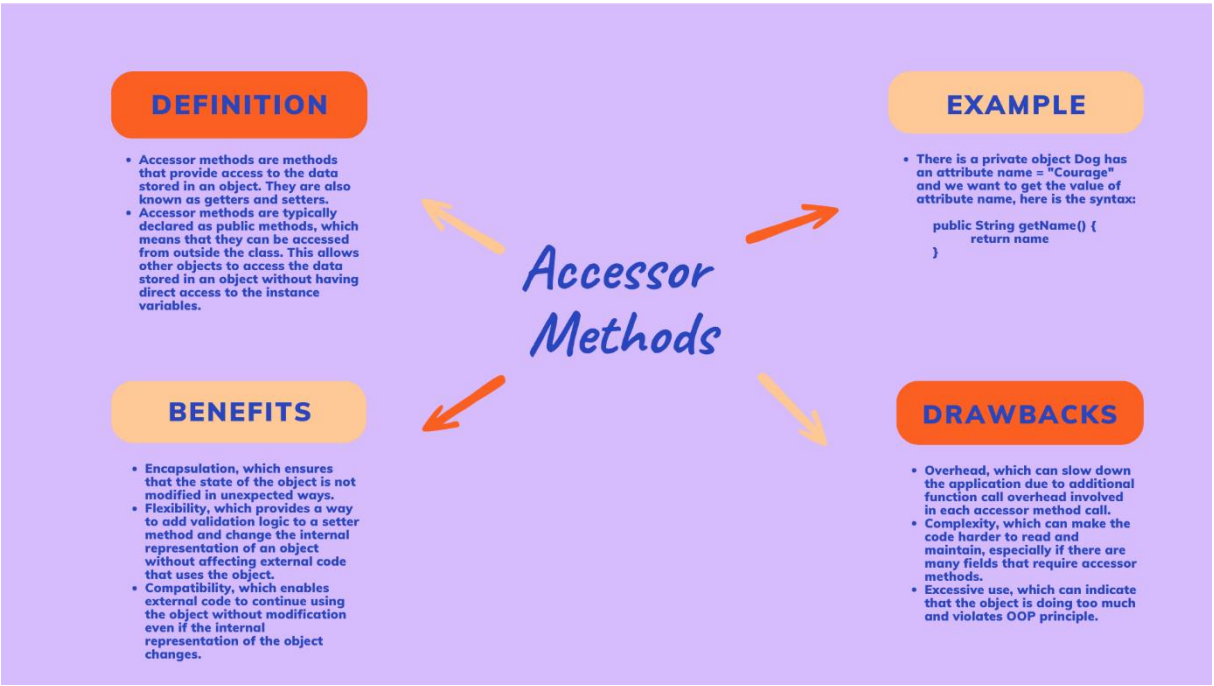
2. Use case diagram



3. Class diagram



4. Reading assignment



Quest: When should accessor methods be used?

Accessor methods, also known as getters and setters, are used to access or modify the values of private instance variables of a class. They should be used when we want to provide controlled access to the private fields of a class while enforcing encapsulation. But we shouldn't use accessor methods unless absolutely necessary

because these methods expose information about how a class is implemented and as a consequence make your code harder to maintain.

Quest: If you create a constructor method to build a DVD by title then create a constructor method to build a DVD by category. Does JAVA allow you to do this?

Since Java only allows us to create multiple constructor methods with the same name but with difference in type of parameters. This is called overloading method. So when we want to create a constructor method to build a DVD by title then create a constructor method to build a DVD by category, but these 2 variables have the same type (both are String)

=> The answer is No.