1. Complexity of Methods

public void displayAll(Student studentArray[]) {

for (int i = 0; i < studentArray.length; i++) { 2n+2

if (studentArray[i] != null) { 1

studentArray[i].printInfo(); 1

}

} Total Cost: 2n+4

} = O(n)

public void searchStudent(String id, Student studentArray[]) {

boolean List = false; 1

for(int i = 0 ; i < studentArray.length ; i++){ 2n+2

if(studentArray[i] != null) { 1

if (studentArray[i].getID() == Integer.parseInt(id)) { 1

studentArray[i].printInfo(); 1

List = true; 1

}

} }

if (!List) {

System.out.print("\nStudent not found"); 1

}

} Total Cost: 2n + 8

= O(n)

1. Classes
2. Person - It gets the inputs of “Name” and “Age” from the MainClass

* It then inherited by the Student Class

1. Student - It gets the input “ID number” from the main class and prints ID#, Name, and Age

* The class extends Person

1. MainClass – it runs the main function of the program

* It will prompt the user to select an input in the User Menu
* Numbers from 1 to 5 has their specific functions
* 1 – adds students to the list
* 2 – removes the student from the list
* 3 – search if a specific student is on the list
* 4 – prints the whole list
* 5 – exit the program

1. StudentMethods – interface implemented by the MainClass

* Methods are:
* displayAll(); - It reiterates the arrayList and prints all existing student in the list
* searchStudent(); - It searches the array to find a specific ID and pints the student info