**CSC272 Advanced Programming in Java**

**Assignment 3**

|  |  |
| --- | --- |
| **First Name** | Asa Jean |
| **Last Name** | Soriano |
| **ID#** | 041702547 |
| **Assignment Week#** | 3 |
| **Email Address** | asajeansoriano@gmail.com |

Contents

[How to submit your Assignment 1](#_Toc52036390)

[Problem 1 – *Generic Selection Sort Method* 2](#_Toc52036391)

[Problem 2 – *Generic Class Pair* 2](#_Toc52036392)

[Problem 3 – *Poly Line App* 2](#_Toc52036393)

# How to submit your Assignment

After filling all the parts in this file, please follow the following steps.

1. Add your name and ID to the first page.
2. Save the file in the original format (Docx or Doc)

(Please **do not** convert to other file formats e.g. PDF, ZIP, RAR, …).

1. Rename the file as

*YOUR* ***First*** *Name - YOUR* ***Last*** *Name- YOUR student ID-* CSC272*.docx*

**Example:**

John – Smith - 234566435 – CSC272.docx

1. Upload the file and submit it (only using Blackboard)

Please do not hesitate to contact me, should you have any questions.

# Problem 1 – *Generic Selection Sort Method*

Write a generic method selectionSort based on the sort program of Fig.19.4 . Write a test program that inputs, sorts and outputs an Integer array and a Float array..

|  |
| --- |
| Your code for this problem |
| public class P1 {  public static void main (String[] args) {  Integer[] intArray = {5, 4, 3, 6, 2, 1, 7, 9, 8, 10};  Float[] floatArray = {4.5f, 6.8f, 2.4f, 1.1f, 1.05f, 3.2f, 5.6f};  for (int i = 0; i < intArray.length; i++) {  System.out.print(intArray[i] + " ");  }  System.out.println();  for (int i = 0; i < floatArray.length; i++) {  System.out.print(floatArray[i] + " ");  }  System.out.println();  selectionSort(intArray);  selectionSort(floatArray);  System.out.println("Selection Sort on integer array");  for (int i = 0; i < intArray.length; i++) {  System.out.print(intArray[i] + " ");  }  System.out.println("\nSelection Sort on float array");  for (int i = 0; i < floatArray.length; i++) {  System.out.print(floatArray[i] + " ");  }  }  //generic method selectionSort that inputs, sorts, and outputs an integer array and a float array  public static <T> void selectionSort(T[] inputArray) {  // loop over inputArray.length - 1 elements  for (int i = 0; i < inputArray.length - 1; i++) {  int smallest = i; // first index of remaining array  // loop to find index of smallest element  for (int index = i + 1; index < inputArray.length; index++) {  T current = inputArray[index];  T smallestValue = inputArray[smallest];  if (current.toString().compareTo(smallestValue.toString()) < 0) {  smallest = index;  }  }    swap(inputArray, i, smallest); // swap smallest element into position  printPass(inputArray, i + 1, smallest); // output pass of algorithm  }  }  // helper method to swap values in two elements  private static <T> void swap(T[] inputArray, int first, int second) {  T temporary = inputArray[first]; // store first in temporary  inputArray[first] = inputArray[second]; // replace first with second  inputArray[second] = temporary; // put temporary in second  }  // print a pass of the algorithm  private static <T> void printPass(T[] inputArray, int pass, int index) {  System.out.printf("after pass %s: ", pass);  // output elements till selected item  for (int i = 0; i < index; i++) {  System.out.printf("%s ", inputArray[i]);  }  System.out.printf("%s\* ", inputArray[index]); // indicate swap  // finish outputting array  for (int i = index + 1; i < inputArray.length; i++) {  System.out.printf("%s ", inputArray[i]);  }  System.out.printf("%n "); // for alignment  // indicate amount of array that�s sorted  for (int j = 0; j < pass; j++) {  System.out.print("-- ");  }  System.out.println();  }  } |

Run the code and insert the result in the following box.

|  |
| --- |
| Sample Run Result |
|  |

# Problem 2 – *Generic Class Pair*

Write a generic class Pair which has two type parameters—F and S—each representing the type of the first and second element of the pair, respectively. Add get and set methods for the first and second elements of the pair.

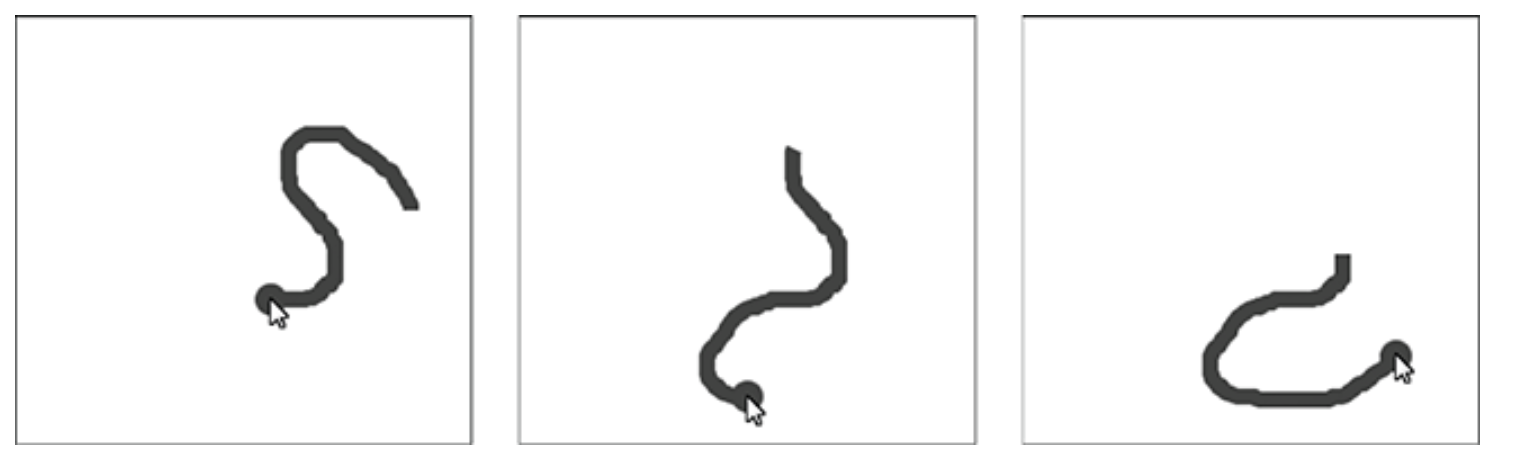
|  |
| --- |
| Your code for this problem |
| public class P2 {  //Generic Class Pair  public static class Pair<T> {  private T first;  private T second;  public Pair(T first, T second) {  this.first = first;  this.second = second;  }  public T getFirst() { return first; }  public T getSecond() { return second; }  public void setFirst(T first) { this.first = first; }  public void setSecond(T second) { this.second = second; }  }  public static void main (String[] args) {  //test Pair class  Pair<Integer> pair1 = new Pair<Integer>(1, 2);  Pair<String> pair2 = new Pair<String>("hello", "world");  Pair<Float> pair3 = new Pair<Float>(1.0f, 2.0f);  System.out.println(pair1.getFirst() + " " + pair1.getSecond());  System.out.println(pair2.getFirst() + " " + pair2.getSecond());  System.out.println(pair3.getFirst() + " " + pair3.getSecond());  //test setFirst and setSecond  pair1.setFirst(3);  pair1.setSecond(4);  pair2.setFirst("hi");  pair2.setSecond("there");  pair3.setFirst(3.0f);  pair3.setSecond(4.0f);  System.out.println(pair1.getFirst() + " " + pair1.getSecond());  System.out.println(pair2.getFirst() + " " + pair2.getSecond());  System.out.println(pair3.getFirst() + " " + pair3.getSecond());  }  } |

Run the code and insert the result in the following box.

|  |
| --- |
| Sample Run Result |
|  |

# Problem 3 – *Poly Line App*

Create an app in which, as the user moves the mouse cursor around the window, a Circle and Polyline follow the cursor. The app should appear as shown in the following figure. There should always be a Circle centered at the current mouse-cursor location, and that location should also be the first point in the Polyline. As you respond to each mouse-move event, use the mouse cursor’s location as the Circle’s new center and insert that location at the *beginning* of the Polyline’s points collection. The Polyline’s length should not increase forever—once the Polyline reaches 50 points, remove the last point each time you insert a new first point.



Circle and Polyline follow the mouse cursor inside the window.

|  |
| --- |
| Your code for this problem |
|  |

Run the code and insert the result in the following box.

|  |
| --- |
| Sample Run Result |
|  |

**The end**