PROG6112 test templates

Table of Contents

Turn document to autosave

Have ready:

- Word;
- Netbeans; and
- GitHub.

Learning units 1-3

PROG6112_test_prep

/* Fetched code from W3 schools for importing scanner class

https://www.w3schools.com/java/java_user_input.asp */

```
import java.util.Scanner;

public class PROG6112_test_prep {

   public static void main(String[] args) {
      // sort(arr);
      // info hiding = att + meth
      // 800*1000

      /*
      arrays are faster, simpler and can store primatives
      arraylists are malleable

      */ /*

      - static = no objects
      - abstract = allows empty meth
```

```
    - NOT for attributes
    - all abstract methods must be defined in child
  - final = constant att + meth
    - OR no subclasses
  -
  - static final = no objects AND constant
  -
  - abstrac & final = OPPOSITES
  */

  // METHODS NEITHER PROMPT INPUT, NOT PRINT OUTPUT !!!!!!!!!!!!!!
}

void Enum() { // create new datatype
  enum directions {north, south, east, west}

  System.out.println(directions.north);

  directions.south.ordinal(); // 1
  directions.south.name(); // "south"
  directions[] x = directions.values(); // array of enum constants
  directions.valueOf("west"); // directions.west
}

void loops(String[][] Serieses, int totSerieses, int totLength) {
  for (String[] rec : Serieses){ // loop through list of serieses
    totSerieses += 1; // increment number of serieses
    totLength += rec.length;
```

```
    }

    boolean found = false;
    String[] ans;
    while (!found) {
      for (String[] rec : Serieses) {
        if (rec[0].equals("hi")) {
          ans = rec;
          found = true;
        }
      }
    }
    if (!found) System.out.println("record was not found");

    /* Fetched code from W3 schools for creating an index-based for loop
    https://www.w3schools.com/java/java_arrays_loop.asp */

    for (int i = 0; i < Serieses.length; i++){ // loop through Serieses index
      //
    }

  }
  void string_int_switching() {
    int x = Integer.parseInt("1"); // convert string to integer

    /* Fetched code from Geeks for Geeks for converting an integer value to a String
    https://www.geeksforgeeks.org/java/different-ways-for-integer-to-string-
conversions-in-java/  */
```

```
    String y = Integer.toString(x); // convert integer to String
  }
  void ErrorHandeling() {
    /* Fetched code from W3 schools for using try-catch-finally block
    https://www.w3schools.com/java/java_try_catch.asp */

    try { // attempt execution of fragile code
      //
    } catch (Exception e) { // catch exception if try block fails
      e.printStackTrace();
    } finally { // always continue script, using the following
      System.out.print("executed");
    }
  }
  void userInput (boolean valid, String ans) {
    while (!valid) {
      /* Fetched code from W3 schools for creating new scanner object
      https://www.w3schools.com/java/java_user_input.asp */

      Scanner Scan = new Scanner(System.in); // declare and initialise new Scanner
object

      /* Fetched code from W3 schools for reading user input
      https://www.w3schools.com/java/java_user_input.asp */

      System.out.println("please enter your answer: "); // prompt user's input
      ans = Scan.nextLine(); // read user input, using Scanner object
```

```
      /* Fetched code from W3 schools for trimming whitespace
      https://www.w3schools.com/java/ref_string_trim.asp */

      ans = ans.trim(); // trim whitespace from user's answer

      valid = ans.equals("");
    }
}
void Switch(int x) {
    /* Fetched code from W3 schools for implementing switch block
    https://www.w3schools.com/java/java_switch.asp */

    switch (x) { // switch to read answer, and execute the correct code
      case 1 : // if answer is 1
        break;
      case 2 : // if answer is 2
        break;
      case 3 : // if asnwer is 3
        break;
      case 4 : // if answer is 4
        break;
      case 5 : // if answer is 5
        break;
      default : // if answer is invalid
        System.out.println("Invalid anser. Please try again.");
    }
```

```
  }
  void Print_F () {
    /* Fetched code from W3 schools for implementing System.out.printf()
    https://www.w3schools.com/java/ref_output_printf.asp */

    System.out.printf("SeriesID: %-15.2f;", 60.0); // display details
  }


//=====================================================================

//=====================================================================

//=====================================================================

  void arrays(int[] arr, int[] arr2) {
    Arrays.sort(arr);
    String a = Arrays.toString(arr); // [, , , ]
    Arrays.fill(arr, 1); // (array, newValue)
    boolean x = Arrays.equals(arr, arr2); // (array1, array2)
    int y = arr.length;
  }

  void arrayList(ArrayList<Integer> arl) {
    arl.add(22);
    int y = arl.get(0);
    arl.remove(0);

    /* Fetched code from W3 schools for updating an ArrayList element
```

```
    https://www.w3schools.com/java/java_arraylist.asp */

    arl.set(1, 100); // (index, newValue)

    /* Fetched code from Geeks for Geeks for reading the length of an ArrayList
    https://www.geeksforgeeks.org/java/java-program-to-find-the-length-size-of-an-arraylist/ */

    int length = arl.size();

    String a = arl.toString(); // [, , , ]
    arl.clear();

    /* Fetched code from Geeks for Geeks for validating if an arraylist is empty
    https://www.geeksforgeeks.org/java/arraylist-isempty-java-example/ */

    arl.isEmpty();
  }

  //================================================================
  //================================================================
  //================================================================

  private static void bubbleSort() { //boolean sorted = false;
    /*
    bubblesort
    - swap pairs
    - cycle to right
```

```
  - the highest/lowest number will naturally be carried to the right
  - repeat cycle
    - slightly shorter
  -
  - if no swaps done IT IS SORTED
  */

  int[] arr = {1, 0, 4, 6, 8, 5, 3, 9};

  /* Fetched code from W3 schools for creating an index-based for loop
  https://www.w3schools.com/java/java_arrays_loop.asp */

  for (int i = 0; i < arr.length -1 ; i++) { // for every element
    boolean sorted = false;
    for (int j = 0; j < arr.length -1 -i ; j++) { // cycle through the list
      if (arr[j] > arr[j+1]) { // if the pair is in descending order
        // swap positions
        int temp = arr[j];
        arr[j] = arr[j+1];
        arr[j+1] = temp;
        sorted = true;
      }
    }
    if (!sorted) break;
  }

  for (int i : arr) System.out.println(i);
}
```

```
static void insertionSort() { // insertion sort practice
  /*
  insertion sort
  - cycle to right
  - when patern-breaking element found
  - save
  - cycle left
    - save elements in previous positions

  insertion sort explanation:
  - save element
  - pull elements across intil position found
  -
  - start with second element
  - compare it with elements before
  - insert it to the correct position
  - repeat for all elements

  for loop + if statement = while ( && ) biconditional loop
  - add in j++ or j--
  */

  int[] arr2 = {1, 0, 4, 6, 8, 5, 3, 9};

  /* Fetched code from W3 schools for creating an index-based for loop
  https://www.w3schools.com/java/java_arrays_loop.asp */
```

```
    for (int i = 1; i < arr2.length -1; i++) { // cycle right - for each element
      int temp = arr2[i]; // save element
      int j = i -1; // start at element behind i

      while (j >= 0 && arr2[j] > temp) { // while still in array && out of order( + NOT same element as i)
        arr2[j +1] = arr2[j]; // shift forward -> j+1 = i
        j--; // ends at j = -1
      }

      // add 1 to j=-1 to indicate last element left off
      arr2[j +1] = temp;
    }

    for (int i : arr2) System.out.println(i);
  }
}
```

Parent

```
public class parent {
   protected final int CATEGORY = 60;
   protected int id;

   protected parent (int id) {
      this.id = id;
      System.out.print(this.id);
   }

   protected void speak() {
      System.out.println("i am an adult");
   }
   protected void walk() {
      System.out.println("i am walking");
   }
}

class child extends parent { // no info hiding
   private final int SUB_CATEGORY = 60;
   private String toy;

   private child (int x, String toy) {
      super(x);
      this.toy = toy;
      System.out.print(super.id);
   }
```

```
    protected void cry() {
        System.out.println("i am sad");
    }

    /* Fetched code from Geeks for Geeks for overriding inherrited methods
    https://www.geeksforgeeks.org/java/overriding-in-java/ */

    @ Override
    protected void speak() {
        super.speak();
        System.out.println("i am a child");
    }
}
```

abstractClass

```
public abstract class abstractClass {
   int childId;


   abstractClass (int id) { // for super () child classes
      this.childId = id;
   }


   protected void standardAct() {
      System.out.println("this is the non-abstract method");
   }
   protected abstract void childAct();
}
```

RandomGenerators

```
// method to generate unique 5-digit string
   private static String createId(ArrayList<String[]> Serieses) {
      String idCode = ""; // declare idCode, for conversion if Id Integer to id String
      String id = ""; // declare ID as a String
      int Id = 0; // declare Integer to increment the ID

      /*
      Fetched code from Geeks for Geeks for reading the length of an ArrayList
      https://www.geeksforgeeks.org/java/java-program-to-find-the-length-size-of-an-arraylist/
      */

      // determines if array is too long
      if (Serieses.size() > 100000) System.out.println("Maximim arraylist length exceeded");

      /* Fetched code from Geeks for Geeks for validating if an arraylist is empty
      https://www.geeksforgeeks.org/java/arraylist-isempty-java-example/  */

      // initialises ID to starting value (00000) if arraylist is empty
      else if (Serieses.isEmpty()) id = "00000";

      // normal ID generation function
      else {
         id = "00000"; // initialise id String
         for (String[] series : Serieses) { // loop through Serieses
            if (series[0].equals(id)) { // the id numbers match - the current id number is already taken
```

```
            Id += 1; // increment by 1


            /* Fetched code from Geeks for Geeks for converting an integer value to a
String
            https://www.geeksforgeeks.org/java/different-ways-for-integer-to-string-
conversions-in-java/  */


            idCode = Integer.toString(Id); // convert ID integer to String


            // add 0s to ID, so there are exactly 5 characters in the ID
            int filler = 5 - idCode.length(); // calculate number of missing characters in
idCode
            id = "0".repeat(filler) + idCode; // concatenate idCode with missing 0s, and set
id String
          } else { // ID doesnt match - current id number does not exist
            break; // stop looping - unique id number has been found
          }
        }
      }

      return id; // return final, unique ID number
    }
```

InterfaceClass

```
interface inter1 { // all are public static - DEFAULT
   final int id = 60;

   void meth1();
}

interface inter2 {
   void meth2();
}

public class InterfaceClass implements inter1, inter2 {
   // implement all OR be declared abstract

   @ Override
   public void meth1() { // must be public
      System.out.println("this is from interface 1");
   }

   @ Override
   public void meth2() { // must be public
      System.out.println("this is from interface 2");
   }
}
```

Template

PROG6112_test_template

```
package prog6112_test_template;

/* Fetched code from W3 schools for importing scanner class
https://www.w3schools.com/java/java_user_input.asp */

import java.util.Scanner;

public class PROG6112_test_template {

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

    String ans = "1";
    boolean exit = false;
    while (!exit) { // repeat code until application is terminated
      //

      exit = !ans.equals("1");
    }
  }

  /* Fetched code from W3 schools for implementing modifiers
  https://www.w3schools.com/java/java_modifiers.asp */
```

```
    private static int meth() {
        return 1;
    }
}

//=========================================================================
===
// END-OF-FILE
//=========================================================================
===
```

SubClass

```
package prog6112_test_template;

class SubClass {
   /* Fetched code from W3 schools for implementing modifiers
   https://www.w3schools.com/java/java_modifiers.asp */

   // attributes
   protected String SeriesId;

   /* Fetched code from Geeks for Geeks for implementing constructors
   https://www.geeksforgeeks.org/java/constructors-in-java/

   Fetched code from W3 schools for implementing modifiers
   https://www.w3schools.com/java/java_modifiers.asp */

   // main constructor
   protected SubClass (String SeriesId) {
      this.SeriesId = SeriesId;
   }

   /* Fetched code from W3 schools for implementing modifiers
   https://www.w3schools.com/java/java_modifiers.asp */

   protected static int meth() {
      return 1;
   }
```

```
}

/* Fetched code from W3 schools for implementing modifiers
https://www.w3schools.com/java/java_modifiers.asp

Fetched code from W3 schools for creating a subclass
https://www.w3schools.com/java/java_inheritance.asp */

class SubSub extends SubClass {
   /* Fetched code from Geeks for Geeks for implementing constructors in subclasses
   https://www.geeksforgeeks.org/java/inheritance-and-constructors-in-java/ */

   // main constructor
   protected SubSub(String SeriesId) {
      super(SeriesId);
   }

}

//=================================================================
==
// END-OF-FILE
//=================================================================
==
```

SubClassTest

```
package prog6112_test_template;

/* Fetched code from Apache NetBeans for creating JUnit tests
https://netbeans.apache.org/tutorial/main/kb/docs/java/junit-intro/#_writing_junit_4_tests */

import org.junit.Test;
import static org.junit.Assert.*;

public class SubClassTest {
   @Test
   public void methTest() {
      System.out.println("SubClassTest : methTest()");

      int expected = 1;
      int actual = SubClass.meth();

      assertEquals(expected, actual);
   }
}

//======================================================================
// END-OF-FILE
//======================================================================
```