**GUIDE QUESTIONS**

**COMPUTER PROGRAMMING II**

**IV.ASSESSMENTS**

1. Write your own version of array copy for int arrays public static void arraycopy(int[] from,int fromstart, int[] to, int tostart, intcount)

**Answer:**

**{ int a[]= {0,1,2,3,4,5,6,7,8,9,10};**

**System.out.print("Array Content:");**

**for(int i=0;i<a.length;i++)**

**{**

**System.out.print(" "+a[i]+", ");**

**}**

1. Write a Java program to print the following grid using array.

**Answer:**

**int a[]= {0,0,0,0,0,0,0,0,0,0};**

**for(int x=0;x<10;x++) {**

**for(int i=0;i<a.length;i++)**

**{**

**System.out.print(" "+a[i]+" ");**

**}**

**System.out.println();**

**}**

1. Write a Java program to calculate the average value of array elements.

**Answer:**

**int[] numbers = new int[]{85, 85, 85, 85, 89, 85, 85};**

**//calculate sum of all array elements**

**double sum = 0;**

**for(int i=0; i < numbers.length ; i++){**

**sum = sum + numbers[i];**

**}**

**//calculate average value**

**double average = sum / numbers.length;**

**System.out.println("Average value of the array elements is : " + average);**

1. Write a Java program to read a string and return true if it ends with a specified string of length 2.

**Answer:**

**{**

**\* To change this license header, choose License Headers in Project Properties.**

**\* To change this template file, choose Tools | Templates**

**\* and open the template in the editor.**

**\*/**

**package angelic\_mod1;**

**import java.util.\*;**

**/\*\***

**\***

**\* @author hp**

**\*/**

**public class Angelic\_mod1 {**

**public boolean endsNg(String str) {**

**int len = str.length();**

**String ng = "lic";**

**if (len < 3) {**

**return false;**

**} else if (ng.equals(str.substring(len - 3, len))) {**

**return true;**

**} else {**

**return false;**

**}**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

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

**// TODO code application logic here**

**Angelic\_mod1 m = new Angelic\_mod1();**

**String str1 = "angelic";**

**System.out.println("The given strings is: " + str1);**

**System.out.println("The string containing ng at last: " + m.endsNg(str1));**

**}**

1. Write a Java program to remove duplicate characters from a given string presents in another given string.

**Answer:**

**String a = "abcde";**

**String b = "abc";**

**System.out.println("The given string is: " + a);**

**System.out.println("The given mask string is: " + b);**

**char arr[] = new char[a.length()];**

**char[] mask = new char[256];**

**for (int i = 0; i < b.length(); i++) {**

**mask[b.charAt(i)]++;**

**}**

**System.out.println("\nThe new string is: ");**

**for (int i = 0; i < a.length(); i++) {**

**if (mask[a.charAt(i)] == 0) {**

**System.out.print(a.charAt(i));**

**}**

**}**

**System.out.println();**

1. Write a Java program to check if a file or directory specified by path name exists or not.

**Answer:**

**File my\_file\_dir = new File("C:\\Users\\hp\\Desktop\\array.txt");**

**if (my\_file\_dir.exists()) {**

**System.out.println("The directory or file exists.\n");**

**} else {**

**System.out.println("The directory or file does not exist.\n");**

**}**

1. Write a Java program to get a list of all file/directory names from the given.

**Answer:**

**File file = new File("C:\\Users\\hp\\Desktop");**

**String[] fileList = file.list();**

**for (String name : fileList) {**

**System.out.println(name);**

**}**

**B. EXPLORE: OTHER CLASS AND METHODS**

1. List down the other class and methods of string, file, and array and table manipulation on the table below.

**FILE**

**TABLE**

**ARRAY**

**STRING**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CLASS** | **METHOD** | **CLASS** | **METHOD** | **CLASS** | **METHOD** | **CLASS** | **METHOD** |
| **1.char charAt**  **(int index)**  **2.Boolean**  **equals(Object obj)**  **3.int Hashcode()**  **4.int compareTO**  **(String string)**  **5.int compareTo**  **IgnoreCase**  **(String string)** | **It returns the character**  **at the specified index. Specified index value**  **should be between 0 to length() -1 both in**  **clusive. It throws IndexOutOfBounds**  **Exception if index<0||>= length of String.**  **Compares the string with the specified string and returns true if both matches else false.**  **It returns the hash code of the string.**  **This method compares the two strings based on the Unicode value of each character in the strings.**  **Same as CompareTo method however it ignores the case during comparison.** | **1.**  **asList()**  **2.** Arrays.binarySearch()3. compare(array 1, array 2)4. compareUnsigned(array 1, array 2) **5. Arrays.copyOf()** | **Acts as bridge between array-based and collection-based APIs, in combination with Collection.toArray(). The returned list is serializable and implements RandomAccess**.  **This method searches a range of the specified array for the specified object using the binary search algorithm.**  **This method compares two arrays passed as parameters lexicographically**.  **This method compares two arrays lexicographically, numerically treating elements as unsigned.**  **This method copies the specified array, truncating or padding with the default value (if necessary) so the copy has the specified length.** | **1.coloumnAdded**  **(TableColumnModelEvent)**  **2.colummoved(TableColumnModelEvent e)**  **3.void columnRemove(TableColumnEvent e)**  **4.void colomnSelectionChanged(ListSelectionEvent e)**  **5.protected voidconfigured EnclosingSscrollPane()** | **Invoked when a column is added to the table column model.**  **Invoked when a column is reposition.**  **Invoked when a coloumn is removed from the table column model.**  **Invoked when the selection model of the TableColumnModel is changed.**  **If this JTableis the viewportView of an enclosing JScroll Pane** | **1.public String getName()**  **2. public String getParent()**  **3.public File getParentFile()**  **4. public String getPath()**  **5. public boolean isAbsolute()** | **Returns the name of the file or directory denoted by this abstract pathname.**  **Returns the pathname string of this abstract pathname's parent, or null if this pathname does not name a parent directory.**  **Returns the abstract pathname of this abstract pathname's parent, or null if this pathname does not name a parent directory.**  **Converts this abstract pathname into a pathname string.**  **Returns the absolute pathname string of this abstract pathname.** |

**D.ELABORATE: Coding**

**Write a java program that uses OOP technique:**

1. Create another class and methods that returning a **charAt()** and **indexOf()** manipulated String.
2. Create another class and methods that convert and return as ArrayList from array data.

**E.EVALUATE: SELF ASSESSMENT**

**IMPROVE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **QUESTIONS** | **YES** | **NO** | **MAYBE** |
| **1. Did I work hard on this module?** |  |  |  |
| **2. Did I understand what my teacher asked me to do?** |  |  |  |
| **3. Did I spend enough time to finish answering this module?** |  |  |  |
| **4. Did I make good use of available resources?** |  |  |  |
| **5. Did I check/ review my work for possible errors?** |  |  |  |
| **5. Did I check/ review my work for possible errors?** |  |  |  |
| **6. Did I learn something in this module?** |  |  |  |
| **7. Did I ask questions if I needed help?** |  |  |  |
| **8. Did I read the instructions carefully?** |  |  |  |
| **9. Did I set high standards for myself?** |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **10. Did I meet the success criteria?** |  |  |  |

**INSTRUCTOR: MR.RODOLFO A. JAVIER Jr.**