|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **V. TEACHING-LEARNING ACTIVITIES** | | | | | | | | | | |
| Note: Add a folder and name it as **module1\_tla** (tla means Teaching Learning Activities) before you commit your answer/document. Commit your answer/document on your remote repository that shared to your instructor github account.  **A. ENGAGE: Reflection**  **Misconception Check**  How the OOP works in arrays, table, string and files manipulation?   * In my own understanding in this subject the OOP works in arrays, table, string and files manipulation in the way of separating the code and put it to other classes to call the methods from that classes so the programmer can organized his/her code for faster way for finding the possible errors or if he/she wants to change the code. | | | | | | | | | | |
| **B. EXPLORE: other class and methods**  1. List down the other class and methods of string, file, array and table manipulation on the table below. | | | | | | | | | | |
|  | | | | | | | | | | |
| STRING | | ARRAY | | TABLE | | | FILE | | | |
| CLASS | METHOD | CLASS | METHOD | CLASS | METHOD | | CLASS | | METHOD | |
| charAt(int index)  compareTo (String another String)  compareTo (Object o)  compareTo IgnoreCase (String str)  concat(String str) | Returns the character at the specified index.  Compares two strings lexico-graphically.  Compares this String to another Object.  Compares two strings lexicograp- hically, ignoring case differences.  Concatenates the specified string to the end of this string. | static int binarySearch (byte[] a, byte  static <T> List<T> asList(T... a)  static int binarySearch (byte[] a, int fromIndex, int toIndex, byte key) | This method searches the specified array of bytes for the specified value using the binary search algorithm.  This method returns a fixed-size list backed by the specified array.  This method searches a range of the specified array of bytes for the specified value using the binary search algorithm. | JTable. Accessible JTable  JTable.Drop Location  JTable.Print Mode  addColumn (TableColumn aColumn)  addNotify() | This class implements accessibility support for the JTable class.  A subclass of Transfer Handler. DropLocation representing a drop location for a JTable.  Printing modes, used in printing JTables.  Appends aColumn to the end of the array of columns held by this JTable's column model.  Calls the configure Enclosing ScrollPane method. | | File(File parent, String child)  File(String pathname)  File(String parent, String child) | | This constructor creates a new File instance from a parent abstract pathname and a child pathname string.  This constructor creates a new File instance by converting the given pathname string into an abstract pathname.  This constructor creates a new File instance from a parent pathname string and a child pathname string. | |
|  | | | | | | | | | | |
|  | | | | | | | | | | |
| **E.EVALUATE**  **Self-Assessment.**  Kindly check (✔) the box of your answer for each question. In this way, we will be able to assess how much we have learned and what are the things that needs to be | | | | | | | | | | |
| improved. | | | | | | | | | | |
| **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?** | | | | | | ✔ | |  | |  |
| **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?** | | | | | | ✔ | |  | |  |