INSTITUTE OF SOFTWARE ENGINEERING

GRADUATE DIPLOMA IN SOFTWARE ENGINEERING

ASSIGNMENT NAME

Programming fundamentals

ASSIGNMENT NO

07

NUMBER OF QUESTIONS: 24

NUMBER OF COMPLETED QUESTIONS: 24

NUMBER OF REMAINING QUESTIONS: 00

STUDENT NAME: M.W.Maheeshi Jayarathna

NIC: 200062000066

BATCH NO: 61

Programming Fundamental
Assignment 07

01. Describe following Java statements (Use diagrams when you need):

a. int[] xr;
b. xr=new int[4];
c. System.out.println(xr);
d. System.out.println(xr[0]); //Explain the output e. xr[0]=100;
f. xr[1]=200;
g. xr[2]=300;
h. xr[3]=400;
i. System.out.println(xr[0]+" "+xr[1]+" "+xr[2] +" "+xr[3]);

a. int[] xr;

 create an array reference/address variable.
 Array name : xr

b. xr=new int[4];

 create an array object(unit).
 Number of elements : 4

c. System.out.println(xr);

 Print addess
 Ex: [I@15db9742

d. System.out.println(xr[0]); //Explain the output

 To access variable, the variable must be initialized. But in this case JVM will assign the default value of 0 automatically when the value is no assigned.
 Print the default value of the element xr[0].
 Xr[0] default value : 0

e. xr[0]=100;

 assign 100 to the variable with the index number 0 in the array object that points from the array reference variable(address) called xr.

f. xr[1]=200;

assign 200 to the variable with the index number 1 in the array object that points from the array reference variable(address) called xr.

g. xr[2]=300;

assign 300 to the variable with the index number 2 in the array object that points from the array reference variable(address) called xr.

h. xr[3]=400;

assign 400 to the variable with the index number 3 in the array object that points from the array reference variable(address) called xr.

i. System.out.println(xr[0]+" "+xr[1]+" "+xr[2] +" "+xr[3]);

Print the value of the element xr[0],xr[1],xr[2] and xr[3].
Print : 100   200   300   400

02. Perform the following tasks for an array called *"fractions"*:

a. Declare a constant ARRAY_SIZE that's initialized to 10.

b. Declare an array with ARRAY_SIZE elements of type double, and initialize the elements to 0.

c. Refer to array element 4.

d. Assign the value 1.667 to array element 9.

e. Assign the value 3.333 to array element 6. f. Sum all the elements of the array, using a "for statement". Declare the integer variable x as a control variable for the loop.

```
a.      final int ARRAY_SIZE = 10;
b.      Double[] fractions = new double[ARRAY_SIZE];
c.      //fractions[4];
d.      fractions[9]=1.667;
e.       fractions[6]=3.333;
f.
        double t=0;
        for(int x=0;x<fractions;x++){
                t=t+fractions[x];
        }
```

03. Write Java statements to perform the following tasks:

a. To create int array to store five integer numbers.

b. To assign integers from the keyboard into the above array without using any iteration.

c. Re-write the question “b” by using a for-loop.

d. To print integers stored in the array without using any iteration.

e. Re-Write the question “d” by using a for-loop.

a.
```
int []n=new int[5];
```

b.
```
System.out.print("input an integer 1 : ");
n[0]=in.nextInt();
System.out.print("input an integer 2 : ");
n[1]=in.nextInt();
System.out.print("input an integer 3 : ");
n[2]=in.nextInt();
System.out.print("input an integer 4 : ");
n[3]=in.nextInt();
System.out.print("input an integer 5 : ");
n[4]=in.nextInt();
```

c.
```
for(int i=0;i<n.length;i++){
        System.out.print("input an integer  “+i+” : “);
        n[i]=in.nextInt();
}
```

d.
```
System.out.print(n[0]+” “+n[1]+” “+n[2]+” “+n[3]+” “+n[4] );
```

e.
```
for(int i=0;i<n.length;i++){
        System.out.print(n[i]+” “);
}
```

04. Write Java statements to accomplish each of the following tasks:

a. Display the value of element 6 of array f.

b. Initialize each of the five elements of one dimensional integer array g to 8.

c. Total the 100 elements of floating-point array c.

d. Copy 11-element array “a” into the first portion of array “b”, which contains 34 elements.

e. Determine and display the smallest and largest values contained in 99-element floating-point array “w”.

a. `System.out.println(f[6]);`

b. `int g[]={8,8,8,8,8};`

c. `double t=0;`

```
         for(int i=0;i<100;i++){
                    t+=c[i];
              }
d.  for(int i=0;i<11;i++){
        b[i]=a[i];
          }
```

```
e.      double min=w[0];
        double max=w[0];
        for(int i=1;i<99;i++){
                    if(min>w[i]){
                         min=w[i];
                    }
                    if(max<w[i]){
                         max=w[i];
                    }
              }
              System.out.println(min);
              System.out.println(max);
```

05. Write Java statements to perform the following tasks:

a. To create an array and store the following numbers by using a single statement
   65, 78, 43, 89, 34, 56, 90, 23, 64, 71, 94, 29

b. To print the size of this array.

c. To print numbers in the array as follows by using traditional for-loop
   [65, 78, 43, 89, 34, 56, 90, 23, 64, 71, 94, 29]

d. To print number in the array as follows by using “for-each” loop
   [65, 78, 43, 89, 34, 56, 90, 23, 64, 71, 94, 29]

e. To print odd numbers.
   [65, 43, 89, 23, 71, 29]

f. To print even numbers.
   [78, 34, 56, 90, 64, 94]

```
a       int[] n={65,78,43,89,34,56,90,23,64,71,94,29};
b       System.out.println(n.length);
c       System.out.print("[ ");
        for(int i=0;i<n.length;i++){
              System.out.print(n[i]+",");
        }
        System.out.println("\b]");

d       System.out.print("[ ");
        for(int i:n){
              System.out.print(i+",");
        }
        System.out.println("\b]");

e       System.out.print("[ ");
        for(int i:n){
                if(i%2==1){
                System.out.print(i+",");
                }
        }
        System.out.println("\b]");
f       System.out.print("[ ");
        for(int i:n){
                if(i%2==0){
                System.out.print(i+",");
                }
        }
        System.out.println("\b]");
```

06. Write Java statements to perform the following tasks:

a. To create an integer array with size 12,
b. To read random numbers (0 to 100), store into the array.
c. To print integers as the following format.
    e.g. [67, 90, 99, 43, 74, 27, 0, 76, 19, 45, 74, 90]
d. To print integers as the reverse format.
    e.g. [90, 74, 45, 19, 76, 0, 27, 74, 43, 99, 90, 67]
e. To find and print the sum of integers.
f. To find and print the maximum of integers.
g. To find and print the minimum of integers.
h. To find and print no of even numbers of the array.
i. To find and print no of odd numbers of the array.
j. To print the integers those hold in the even indexes (0, 2, 4, 6, . . .)
k. To print the integers those hold in the odd indexes (1, 3, 5,7, . . .)

```
a.  int[] ar= new int[12];
b.  System.out.print ("[");
    for(int i=0 ; i<12;i++){
       ar[i]=r.nextInt(101);
       System.out.print(ar[i]+", ");
     }
    System.out.println("\b\b]");
c.  System.out.print ("[");
    for(int i=11 ; i>0;i--){
       System.out.print(ar[i]+", ");
     }
    System.out.println("\b]");
    int sum=0;
```

```
for (int i : ar) {
 sum += i;
}
 System.out.println(sum);
 int max = ar[0];
 for (int i = 0 ; i max ){
max = ar[i];
System.out.print("Maximum number :" + max );
}
}
```

07. Using the code fragment given bellow write the Java statement to perform following task:

```
int[] ar={1, 2, 3, 4, 5};
 int[] br=new int[]{10, 20, 30, 40, 50};
```

a. To print data of “**ar”**. //[1, 2, 3, 4, 5]
b. To increment elements of the array “**ar**” by 1.
c. To print array “**ar**” using method “toString” of class “Arrays”. //[[2, 3, 4, 5, 6]
d. To check and print whether both arrays are the same size or not. //”Both arrays are the same size”
e. To add each element of **br** to each element of **ar**. // [12, 23, 34, 45, 56]
f. To copy each value of **br** to **ar.** //[10, 20, 30, 40, 50]

```
import java.util.*;
class Example{
    public static void main(String args[]){
        int[] ar={1, 2, 3, 4, 5};
        int[] br=new int[]{10, 20, 30, 40, 50};
        int[] cr=new int[ar.length];

       for(int i=0;i<ar.lenght;i++){
              System.out.print(ar[i]+", ");
       }
       System.out.println();
      Arrays.myMethod();
       System.out.println();

       for(int i=0;i<ar.lenght;i++){
              ar[i]=br[i];
              System.out.print(ar[i]+", ");
        }
    }
}

 class Arrays{
```

```
public static void myMethod(){
    int[] ar={1, 2, 3, 4, 5};
    for(int i=0;i<ar.length;i++){
        ar[i]++;
        System.out.print(ar[i]+", ");
     }

     int[] br=new int[]{10, 20, 30, 40, 50};
     int[] cr=new int[ar.length];
     System.out.println();
      for(int i=0;i<cr.length;i++){
          cr[i]=ar[i]+br[i];
          System.out.print(cr[i]+", ");
      }
   }
}
```

08. The following program is designed to analyse the performance of a teacher who is teaching mathematics in several classes. The program should be able to input no of students of a particular class and after input subject marks of each student. You are given to complete this program to get the final results.

```
import java.util.*;
class Example{
   public static void main(String args[]){
                        Scanner input=new Scanner(System.in);

                                        System.out.print("Input no of students : ");
      final int N=input.nextInt();

      //1. Create an array to store student marks

      //2. Input marks from the keyboard

      //3. find total
      int total=0;

      //4. find max;

      //5. find min

      //6. print marks [32, 45, 54, 76, ...]

      System.out.println("Total : "+total);
                        System.out.println("Maximum : "+max);
```

```
                                  System.out.println("Minimum : "+min);
          }
      }
```

1.
   ```
   int[] marks=new int[N];
   ```

2.
   ```
   for(int i=0;i<N;i++){
       System.out.println("Input Marks"+(i+1)+":");
       marks[i]=input.nextInt();
    }
   ```

3.
   ```
   int total=0;
     for(int i=0;i <N;i++){
       total+=marks[i];
    }
   ```

4.
   ```
   int max=marks[0];
     for(int i=1;i<N;i++);
        if(marks[i]>N);
    }
   ```

5.
   ```
   int min=marks[0];
     for(int i=1;i<N;i++);
        if(marks[i]<N);
    }
   ```

6.
   ```
   System.out.print("[");
    for(int i=0;i<N;i++);
      System.out.print(marks[i]+", ");
   }
   ```

09. Complete above program (Question 8) by using Java methods

a. Create an array to store student marks
```
int[] marks=createAnArray(N);
```

b. Input marks from the keyboard
```
readMarks(marks);
```

c. Find total
```
int total=total(marks);
```

d. Find max;
```
int max=max(marks);
```

e. Find min

```
    int min=min(marks);
```

f. Print marks [32, 45, 54, 76, ...]

```
    printMarks(marks);
```

a.
```
public static int[] createAnArray(int noOfStudent) {
    int[] array = new int[noOfStudent];
        return array;
}
```

b.
```
public static void readMarks(int[] array) {
    Scanner input = new Scanner(System.in);
    for(int i = 0; i < array.length; i++) {
        System.out.println("Input mark " + (i+1) + ": ");
        array[i] = input.nextInt();
     }
}
```

c.
```
public static int total(int[] array) {
     int total = 0; for(int i = 0; i < array.length; i++) {
     total += array[i];
     }
     return total;
}
```

d.
```
public static int max(int[] array) {
    int max = 0; for(int i = 0; i < array.length; i++) {
     if(array[i] > max) {
          max = array[i];
    }
 }
 return max;
}
```

e.
```
public static int min(int[] array) {
     int min = array[0];
     for(int i = 1; i < array.length; i++) {
         if(array[i] < min) { min = array[i];
         }
    }
    return min;
}
```

f.
```
public static void printMarks(int[] array) {
    System.out.print("[");
    for (int i = 0; i < array.length; i++) {
        System.out.print(array[i] + " ");
     }
     System.out.println("\b]");
}
```

10. Which the following are correct array declaration:

A. int[] a;
B. int []b;
C. int e[5];
D. int c[];
E. int [d];

- A , B , D

11. Which the following are correct array memory allocation:

A. int[] a=new int[5];
B. int[] b=new int[];
C. int[] c=[10, 20, 30, 40, 50]
D. int[] d={10, 20, 30, 40, 50}
E. int[] e=new int[]{10, 20, 30, 40, 50}
F. int[] f=new int[5]{10, 20, 30, 40, 50}
G. int[] g=new int[0];

- A , D , E , G

12. Which can be insert at line 12, still code will compile? class Example{

```
    public static void main(String args[]){
            int[] array;
           //Insert code here //Line 12
    }
}
```

A. array=new int[5];
B. array=new int[10];
C. array=new int[-5];
D. array={10, 20, 30, 40, 50};
E. array=new int[]{10, 20, 30, 40, 50};
F. array=new int[]{};

- A , B , E

13. What are the default values each data type? Demonstrate your answer by using appropriate examples?

```
class Example{
    public static void main(String args[]){
        byte[] ar=new byte[3];
        short[] br=new short[3];
        int[] cr=new int[3];
        long[] dr=new long[3];
        float[] er=new float[3];
        double[]fr=new double[3];
        char[] gr=new char[3];
        boolean[] hr=new boolean[3];

        String[] names=new String[3];

       System.out.println("byte : "+ar[0]+" "+ar[1]+" "+ar[2]);
       System.out.println("short : "+br[0]+" "+br[1]+" "+br[2]);
       System.out.println("int : "+cr[0]+" "+cr[1]+" "+cr[2]);
       System.out.println("long : "+dr[0]+" "+dr[1]+" "+dr[2]);
       System.out.println("float : "+er[0]+" "+er[1]+" "+er[2]);
       System.out.println("double : "+fr[0]+" "+fr[1]+" "+fr[2]);
       System.out.println("char : "+gr[0]+" "+gr[1]+" "+gr[2]);
       System.out.println("boolean: "+hr[0]+" "+hr[1]+" "+hr[2]);
       System.out.println("String : "+names[0]+" "+names[1]+" "+names[2]);
     }
 }
```

14. Which one of the following to get the length of given array:

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

A. array.length();

B. array.length;

C. array.size();

D. array.size;

E. array.length-1;

- B

15. Which the following are legal Java statements: Explain your answer.

A. Int a=new int[10];

B. int b=new int[10].length;

C. int c={10,20,30,40}.length;

D. int d=new int[]{10,20,30,40}.length;

E. int e=new double[]{1.1, 1,2, 1,5, 1,4}.length;

F. int f=new int[]{10,20,30,40}[2];

G. int[] g=new int[]{10,20,30,40}[2];

H. int h=new double[]{1.1, 1,2, 1,5, 1,4}.[2];

I. int i=new double[]{1.1, 1,2, 1,5, 1,4}[2];

J. double j=new double[]{1.1, 1,2, 1,5, 1,4}.[2];

- A , B , D , E , F , G , J

16. Which can be insert at line 10, still code will compile? class Example{

```
    public static void main(String args[]){
        //Insert code here //Line10
        int[] marks=new int[a];
    }
}
```

A. byte a=10; B. short a=10;
C. int a=10; D. long a=10;
E. float a=10; F. double a=10;
G. char a='A'; H. int[] a=new int[10];

- A , B , C , G

17. What is the output? Briefly explain your answer: class Example{

```
    public static void main(String args[]){
    public static void increment(int x, int[] y){
        x++;
        y[0]++;
    }
    public static void main(String args[]){
        int x=100;
        int[] y={200};
        System.out.println(x+" "+y[0]); //print 200
        increment(x,y);
        System.out.println(x+" "+y[0]); //print 201
    }
}
```

18. Define Java method “merge(….)” to complete the following program to get the relevant output. class Example{

```
    public static void main(String args[]){
        char[] vowels1={'a','e','i','o','u'};
        char[] vowels2={'A','E','I','O','U'};
              System.out.println(Arrays.toString(vowels1));  //[a, e, i, o, u]
             System.out.println(Arrays.toString(vowels2));  //[A, E, I, O, U]
```

```
                                             char[] vowelsAll=merge (vowels1,vowels2);
              System.out.println(Arrays.toString(vowelsAll)); //[a, e, i, o, u, A, E, I, O,
                                                                                       U]
       }
    }
```

```
import java.util.Arrays;
class Example{
    public static void main(String args[]){
          char[] vowels1={'a','e','i','o','u'};
          char[] vowels2={'A','E','I','O','U'};

          System.out.println(Arrays.toString(vowels1)); //[a, e, i, o, u]
System.out.println(Arrays.toString(vowels2)); //[A, E, I, O, U]

          char[] vowelsAll=merge (vowels1,vowels2);

System.out.println(Arrays.toString(vowelsAll));//[a, e, i, o, u, A, E, I, O, U]
 }
     public static char[] merge(char[] v1,char[] v2){
            char[] mergedVowels=new char[v1.length+v2.length];

            for(int i=0;i<v1.lenght;i++){
                mergedVowels[i]=v1[i];
            }
            for(int i=0;i<v2.length;i++){
                 mergedVowels[i+v1.length]=v2[i];
            }
            return mergedVowels;
      }
}
```

19. What is the output? Explain your answer. import java.util.*;

```
      class Example{
                           public static void main(String args[]){
            int[] array={100, 200, 300};
                                        System.out.println(Arrays.toString(array));
            for(int a : array){a++;}
                                        System.out.println(Arrays.toString(array));
                                      for(int i=0; i<array.length;i++){array[i]++;}
                                        System.out.println(Arrays.toString(array));
         }
      }
```

```
            [100, 200, 300]    // for(int a : array){
```

```
                        a++
                    }

[100, 200, 300]    //  "each loop" is read only loop, it can't modify and increment
                        for(int i=0;i<array.length;i++){
                              array[i]++;
                         }

[101, 201, 301]    //  Traditional "for loop" we can modify and increment
```

20. Which can be inserted at line 10, still code will compile?

```
import java.util.*;
class Example{
   public static void printArray(int[] array){
      //body
   }
   public static void main(String args[]){
      int[] a=new int[10];
      int[] b=new int[]{10,20,30,40};
      int[] c={10,20,30,40};
      int[] d={};
      //Insert code here //Line 10

   }
}
```

A. printArray(a); B. printArray(b);
C. printArray(c); D. printArray(d);
E. printArray(10,20,30,40); F. printArray({});
G. printArray(new int[]{}); H. printArray(new int[5]);
I. printArray([10,20,30,40]); J. printArray({10,20,30,40});
K. printArray(new int[]{10,20,30,40});

- A , B , C , D , G , J

21. Which can be insert at line 20, still code will compile? class Example{

```
   public static void main(String args[]){
      int x=0;
      int[] xr=new int[3];
      double d=0.0;
      double[] dr=new double[5];
      int[] grade={'a','b'};
      //Insert code here //Line 20
   }
```

```
}
```

A. x=xr[0]; B. xr[0]=x; C. x=xr;
D. xr=x; E. dr[0]=xr[0]; F. xr[0]=dr[0];
G. xr[0]=(int)dr[0]; H. xr=dr; I. dr=(double[])xr;
J. dr=xr; K. xr=(int)dr; L. xr=(int[])dr;

- A , B , E , G

22. Write an application that inputs five numbers, each between 10 and 100, inclusive. As each number is read, display it only if it's not a duplicate of a number already read. Provide for the "worst case," in which all five numbers are different. Use the smallest possible array to solve this problem. Display the complete set of unique values input after the user enters each new value. Use a one-dimensional array to solve the above problem.

```
import java.util.*;
import java.util.Arrays;
class Example{
     public static boolean searchElement(int[] ar, int num){
          for(int i=0;i<ar.length;i++){
              if(ar[i]==num){
                   return true;
              }
          }
          return false;
     }

      public static int[] pushToArray(int[] ar, int num){
         int[] tempArray=new int[ar.length+1];
         for(int i=0;i<ar.length;i++){
              tempArray[i]=ar[i];
         }
         tempArray[tempArray.length-1]=num;
          return tempArray;
      }

      public static void main(String args[]){
          Scanner input=new Scanner(System.in);
          System.out.print(”Input an integer(press -1 to terminate):”);
          int num=input.nextInt();
          int[] ar=new int[0];
          while(num!=-1){
                 boolean isDuplicate=searchElement(ar,num);
                 if(isDuplicate){
                         System.out.println(num+”is duplicate !. re-enter…”);
                  }else{
                          ar=pushToArray(ar,num);
                  }
```

```
            System.out.print(”Input an integer(press -1 to terminate):”);
            num=input.nextInt();
        }
        System.out.print(Arrays.toString(ar));
      }
    }
```

23. Write a method called copyRange that takes as parameters two arrays a1 and a2, two starting indexes i1 and i2, and a length l, and copies the first l elements of a1 starting at index i1 into array a2 starting at index i2. Assume that the arguments' values are valid, that the arrays are large enough to hold the data, and so on.

24. Create a Java application to maintain a number as a **LIST**. You may use an integer array to store integer data and all number should be positive and zeros are not allowed to be input. 0's in the array is counted as no numbers input. This size of the list is equal to the no. of numbers already stored into the array. Following features (functions) should be implemented to complete the application.

a. Create a method *insert(int[] array, int number)*, which accepts two arguments (array, number). "*array*" is the created array and "*number"* is what you insert into it. Note that the given size of the array should increase if the count exceeds the size of the array.

b. Create a method called "printList(int[] array)" to print all numbers in the LIST as follows:

[23, 54, 46, 72, 72, 12, 62, 11, 92. . . . .]

c. Create a method called "*remove()"*, which deletes the last number input in the array.

d. You should overload the method created in the exercise **b** called "*remove(int index)*", "index" is the location which number will be deleted. Note that the "index" should be in between indexes already value in the array. e. Overload the method exercise **a** called

"*insert(int array, int number, int index)"*, here "index" is the location where the number should be inserted. (Apply all validations)

f. Create a method called "*size"* to get the size of the LIST (public static int size(int array).

g. Create a method called "*isEmpty()"* to check the LIST is empty.
(public static Boolean isEmpty(int[] aarray))

h. Create a method called "*isFull()",* to check the LIST is empty.
(public static boolean isFull(int[] array) i. Create a method called "*clear() ",* to clear the LIST.
(public static void clear(int[] array) all elements should be zero.

j. Create a method called “*removeDuplicates()*” to remove all
duplicates numbers of the LIST. k. Create a method called
“*searchElement()*” to find the location of a given number.

```
public static int search(int[] array, int number)
```

l. Create a method called search() to check whether a given number exists if the LIST.

```
public static boolean isExist(int[] array, int number)
```