

## Methods – Parameters Exercise

1. Create a method called print that receives an array of doubles. This method will print all the elements of the array.

```
double [] array = {1.1, 2.2, 3.3, 4.4, 5.5};  
Print (array);
```

```
    }  
    public static void Print (double [] array) {  
        for (int x=0; x<array.length; x++) {  
            System.out.println(array[x]);  
        }  
    }  
}
```

2. Create a method called Largest that receives an array of doubles. The method will return the largest element in the array.

```
double [] array = {1.1, 2.2, 3.3, 4.4, 5.5};  
System.out.println(Largest(array));
```

```
    }  
    public static double Largest (double [] array) {  
        double largest=0;  
        for (int x=0; x<array.length; x++) {  
            if (largest<array[x]) {  
                largest = array[x];  
            }  
        }  
        return largest;  
    }  
}
```

3. Create a method called Smallest that receives an array of doubles. The method will return the smallest positive element in the array.

```
double [] array = {1.1, 2.2, 3.3, 4.4, 5.5};  
System.out.println(Smallest(array));
```

```
    }  
    public static double Smallest (double [] array) {  
        double smallest=array[1];  
        for (int x=0; x<array.length-1; x++) {  
            if (smallest>array[x]) {  
                smallest = array[x];  
            }  
        }  
        return smallest;  
    }  
}
```

4. Create a method called Sum that receives an array of doubles. This method will return the sum of all the number in the array.

```
double [] array = {5.5, 7.5, 1.5, 3.5, 4.5};  
System.out.print(Sum(array));
```

```

    }
    public static double Sum (double [] array) {
        double sum=0;
        for (int x=0; x<array.length; x=x+1) {
            sum = sum +array[x];
        }
        return sum;
    }
}

```

5. Create a method called Average that receives an array of doubles. This method will return the average of all the positive elements in the array.

```

double [] array = {5.5, -7.5, 1.5, 3.5, 4.5};
System.out.println(Average(array));

}
public static double Average (double [] array) {
    double sum=0;
    double count=0;
    double average=0;
    for (int x=0; x<array.length; x++) {
        if (array[x]>=0) {
            sum = sum + array[x];
            count++;
        }
    }
    average = sum/count;
    return average;
}
}

```

6. Create a method called Mode that receives an array of doubles. This method will return the positive value that occurs the most in the array. If there is no value that occurs the most, this method will return -1.

```

double [] array = {1.0,2.0,3.0,4.0,-3.0,4.0};
System.out.println(Mode(array));

}
public static double Mode (double [] array) {
    double mode=0;
    double count =0;
    double maxCount = 0;
    for (int x=0; x<array.length; x++) {
        for (int y=x+1; y<array.length; y++) {
            if (array[x] == array[y] && array[x]>=0) {
                count++;
                if (count>maxCount) {
                    maxCount = count;
                    mode = array[x];
                    count=0;
                }
            }
        }
    }
    return mode;
}
}
}

```

```

        if (maxCount==0) {
            return -1;
        }
        return mode;
    }
}

```

7. Create a method called Median that receives an array of doubles. This method will return the median value of all positive values in the array. That is, for the set of numbers in the array, half should be above the median and half should be below.

```

double [] array= {1,2,3,4,5,6,-3};
System.out.println(Median(array));

}
public static double Median (double [] array) {
    double median = 0;
    double [] temp = new double [array.length-1];
    for (int x=0; x<array.length; x++) {
        if (array[x]>=0) {
            temp[x] = array[x];
        }
    }
    array = temp;

    int length = array.length;
    if (length%2==1) {
        median = array[length/2];
    }
    else if (length%2==0) {
        median = (array[length/2] + array[(length/2)-1])/2;
    }
    return median;
}

```

8. Create a method called Reverse that receives an array of integers. This method will copy the indices in reverse order in a separate array. This method will return the array. –

```

double [] array = {1,2,3,4,5,6};
Reverse(array);

}
public static void Reverse (double[] array) {
    double length = array.length-1;
    double [] reverse = new double [array.length];
    for (int x=0; x<array.length; x++) {
        reverse[(int) length]= array[x];
        length--;
    }

    for (int x=0; x<reverse.length; x++) {
        System.out.println(reverse[x]);
    }
}
}

```

9. Write a method that will keep track of a set of student marks that are stored in array.  
This program will offer the user the following options and print the following

- a. Mean Average
- b. Lowest Mark
- c. Highest Mark
- d. Median
- e. Mode

```
Scanner input = new Scanner (System.in);
System.out.println("Enter number of student:");
int studentNumber = input.nextInt();
double studentMark =0;
double [] mark = new double [studentNumber+1];
for (int x=1; x<=studentNumber; x++) {
    System.out.println ("Enter mark of student " + x);
    studentMark = input.nextDouble();
    mark[x] = studentMark;
}
Option();
int option = input.nextInt();
do {
    if (option==1) {
        System.out.println("The mean average is " +
MeanAverage(mark));
    }
    else if (option==2) {
        System.out.println("The lowest mark is " +
LowestMark(mark));
    }
    else if (option==3) {
        System.out.println("The highest mark is " +
HighestMark(mark));
    }
    else if (option==4) {
        System.out.println("The median is " + Median(mark));
    }
    else if (option==5) {
        System.out.println("The mode is " + Mode(mark));
    }
    else {
        System.out.println("Error, invalid option");
    }
}
while (option>5);
}
public static void Option () {
    System.out.println(" *OPTIONS*");
    System.out.println("1 - Mean Average");
    System.out.println("2 - Lowest Mark");
    System.out.println("3 - Highest Mark");
    System.out.println("4 - Median");
    System.out.println("5 - Mode");
    System.out.println("Enter option:");
}
```

```

}

public static double MeanAverage (double [] mark) {
    double sum = 0;
    double length = mark.length-1;
    for (int x=1; x<=length; x++) {
        sum=sum + mark[x];
    }
    double meanAverage = sum/length;
    return meanAverage;
}

public static double LowestMark (double [] mark) {
    double lowest = mark[1];
    for (int x=1; x<=mark.length-1; x++) {
        if (mark[x]<lowest) {
            lowest=mark[x];
        }
    }
    return lowest;
}

public static double HighestMark (double [] mark) {
    double highest = mark[1];
    for (int x=1; x<=mark.length-1; x++) {
        if (mark[x]>highest) {
            highest=mark[x];
        }
    }
    return highest;
}

public static double Median (double [] mark) {
    double median = 0;
    int length = mark.length-1;
    if (length%2==1) {
        median = mark[(length+1)/2];
    }
    else if (length%2==0) {
        median = (mark[(length)/2] + mark[(length/2)+1])/2;
    }
    return median;
}

public static double Mode (double [] mark) {
    double count = 0;
    double temp = 0;
    double mode = 0;
    for (int x=1; x<=mark.length-1; x++) {
        for (int y=x+1; y<mark.length; y++) {
            if (mark[x]==mark[y]) {
                count++;
            }
            if (temp<count) {
                mode = mark[x];
            }
        }
    }
}

```

```
        }  
        temp=count;  
    }  
}  
if (count==0) {  
    return -1;  
}  
return mode;  
}
```