1. Why Array starts from 0?

The array index in C starts with 0 because in C the name of an array is a pointer, which is a reference to a memory location. Therefore, an expression \*(arr + n) or arr[n] locates an element n-locations away from the starting location because the index is used as an offset. Likewise, the first element of the array is exactly contained by the memory location that array refers (0 elements away), so it should be denoted as \*(arr + 0) or \*(arr) or arr[0].

1. Difference between arguments and parameters?

|  |  |
| --- | --- |
| **ARGUMENT** | **PARAMETER** |
| When a function is called, the values that are passed in the call are called arguments. | The values which are written at the time of the function prototype and the definition of the function. |
| These are used in function call statement to send value from the calling function to the called function. | These are used in function header of the called function to receive the value from the arguments. |
| During the time of call each argument is always assigned to the parameter in the function definition. | Parameters are local variables which are assigned value of the arguments when the function is called |
| They are also called Actual Parameters | They are also called Formal Parameters |

1. Why private is not allowed with classes?

As we already know a field defined in a class using private keyword can only be accessible within the same class and is not visible to outside world.  
  
Let’s consider below example of class A

**package** com.example;

class A {

**private** **int** a = 10;

*// We can access a private field by creating object of same class inside the same class*

*// But realy no body creates object of a class inside the same class*

**public** **void** **usePrivateField**(){

A objA = **new** A();

System.out.println(objA.a);

}

}

Field ‘a’ is declared as private inside ‘A’ class and because of it ‘a’ field becomes private to class ‘A' and can only be accessed within ‘A’. Now let’s assume we are allowed to declare class ‘A’ as private, so in this case class ‘A’ will become private to package ‘com.example’ and will not be accessible from outside of the package.  
  
So defining private access to the class will make it accessible inside the same package which default keyword already do for us, Therefore there is no benefit of defining a class private it will only make things ambiguous.

1. Write a program for Prime Number

**public** **class** PrimeNumber {

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

// **TODO** Auto-generated method stub

**int** num=11,flag=0;

**if**(num == 0 || num == 1)

{

System.***out***.println(num+" No. is not Prime");

}

**else**

{

**for**(**int** i=2;i<num/2;i++)

{

**if**(num%i==0)

{

System.***out***.println(num+" No. is not Prime");

flag=1;

**break**;

}

}

**if**(flag==0)

{

System.***out***.println(num+" No. is Prime");

}

}

}

}

1. Write a program Fibonacci code 0,1,1,2,3,5,8,13,21,34 till 10th place

**public** **class** Fibonacciseries {

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

// **TODO** Auto-generated method stub

**int** n1=0, n2=1,n3,count=10;

System.***out***.print(n1+" "+n2);

**for**(**int** i=2;i<count;i++)//loop starts from 2 b’xoz 0 & 1 are already printed

{

n3=n1+n2;

System.***out***.print(" "+n3);

n1=n2;

n2=n3;

}

}

}

1. Write a program How to find maximum/minimum value in array

**public** **class** MaxAndMin {

**public** **int** max(**int**[] array)

{

**int** max=0;

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

{

**if**(array[i]>max)

{

max=array[i];

}

}

**return** max;

}

**public** **int** min(**int**[] array)

{

**int** min=array[0];

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

{

**if**(array[0]<min)

{

min=array[i];

}

}

**return** min;

}

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

// **TODO** Auto-generated method stub

**int**[] myArray={10,20,30,40,50};

MaxAndMin m=**new** MaxAndMin();

System.***out***.println("Maximum value in the array is: "+m.max(myArray));

System.***out***.println("Minimum value in the array is: "+m.min(myArray));

}

}

Output:

Maximum value in the array is: 50

Minimum value in the array is: 10

1. Write a program to sort array

**public** **class** SortAsc {

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

// **TODO** Auto-generated method stub

//Initialize array

**int**[] arr={15,10,25,20,35,30};

**int** temp = 0;

//Displaying elements of original array

System.***out***.println("Elements of original array:");

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

{

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

}

//Sort the array in ascending order

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

{

**for**(**int** j=i+1;j<arr.length;j++)

{

**if**(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

//Displaying elements of array after sorting

System.***out***.println("\n\nElements of array sorted in ascending order: ");

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

{

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

}

}

}

Output:

Elements of original array:

15 10 25 20 35 30

Elements of array sorted in ascending order:

10 15 20 25 30 35