**Using a Variable or Array Element That Is Uninitialized and Unassigned**

1.Given the following code what will element *b[5]* contain?

public class MyVal{

public static void main(String argv[]){

MyVal m = new MyVal();

m.amethod();

}

public void amethod(){

boolean b[] = new boolean[5];

}

}

1. 0  
   2) null  
   3) ""  
   4) none of these options

Ans:

4) none of these options

Sneaky one here. Array element numbering starts at 0, therefore there is no element 5 for this array. If you were to attempt to perform

System.out.println(b[5])

You would get an exception.

**2)**

Given the following constructor what will element 1 of *mycon* contain?

MyCon(){

int[] mycon= new int[5];

}

1. 0  
   2) null  
   3) ""  
   4) None of these options

Ans:

1) 0

A constructor acts no different to any other method for this purpose and an array of integers will be initialised to contain zeros wherever it is created.

#### 3)

What will happen when you attempt to compile and run the following code?

public class MyField{

int i=99;

public static void main(String argv[]){

MyField m = new MyField();

m.amethod();

}

void amethod(){

int i;

System.out.println(i);

}

}

1) The value 99 will be output  
2) The value 0 will be output  
3) Compile time error  
4) Run time error

#### Ans

3) Compile time error

You will get a compile time error indicating that variable *i* may not have been initialised. The classs level variable *i* is a red herring, as it will be shadowed by the method level version. Method level variables do not get any default initialisation.

#### 4)

What will happen when you attempt to compile and run the following code?

public class MyField{

String s;

public static void main(String argv[]){

MyField m = new MyField();

m.amethod();

}

void amethod(){

System.out.println(s);

}

}

1. Compile time error s has not been initialised  
   2) Runtime error s has not been initialised  
   3) Blank output  
   4) Output of null

Ans

4) Output of null

A variable created at class level will always be given a default value. The default value of an object reference is *null* and the *toString* method implicitly called via System.out.println will output *null*

5. // Java program to demonstrate default values of array

// elements

class ArrayDemo

{

public static void main(String[] args)

{

System.out.println("String array default values:");

String str[] = new String[5];

for (String s : str)

System.out.print(s + " ");

System.out.println("\n\nInteger array default values:");

int num[] = new int[5];

for (int val : num)

System.out.print(val + " ");

System.out.println("\n\nDouble array default values:");

double dnum[] = new double[5];

for (double val : dnum)

System.out.print(val + " ");

System.out.println("\n\nBoolean array default values:");

boolean bnum[] = new boolean[5];

for (boolean val : bnum)

System.out.print(val + " ");

System.out.println("\n\nReference Array default values:");

ArrayDemo ademo[] = new ArrayDemo[5];

for (ArrayDemo val : ademo)

System.out.print(val + " ");

}

}

Output:

String array default values:

null null null null null

Integer array default values:

0 0 0 0 0

Double array default values:

0.0 0.0 0.0 0.0 0.0

Boolean array default values:

false false false false false

Reference Array default values:

null null null null null

**MCQ’s**

**1) An Array in Java is a collection of elements of \_\_\_ data type.**

A) Same

B) Different

C) -

D) -

Answer [=]

**A**

**2) The Java Virtual Machine (JVM) implements arrays as \_\_\_ type.**

A) Primitive

B) Object

C) -

D) -

Answer [=]

**B**

**Explanation:**

**That is the reason why Java Array has predefined methods.**

**3) Unlike C-Arrays, the Java-Arrays have \_\_\_.**

A) Names

B) Values

C) Methods and Fields

D) None

Answer [=]

**C**

**4) An array declaration in Java without initialization \_\_\_ memory.**

A) Does not allocate

B) Allocates memory

C) -

D) -

Answer [=]

**A**

**Explanation:**

**Only initialization causes memory to be allocated.**

**5) In Java language, an array index starts with \_\_\_.**

A) -1

B) 0

C) 1

D) Any integer

Answer [=]

**B**

**6) Which are the special symbols used to declare an array in Java?**

A) Braces { }

B) Parentheses ()

C) Square Brackets [ ]

D) Angled Brackets < >

Answer [=]

**C**

**7) Which are the special symbols used to initialize an array at the time of the declaration itself?**

A) Parentheses ( )

B) Square Brackets [ ]

C) Braces { }

D) Angled Brackets < >

Answer [=]

**C**

**Explanation:**

**int[] nums = {1,3,6};**

**8) It is possible to skip initializing some elements of the array during Shorthand Initialization. (TRUE / FALSE)**

A) FALSE

B) TRUE

C) -

D) -

Answer [=]

**A**

**Explanation:**

**No, you can not skip any elements. All elements need to be initialized in one go or at the same time.**

**9) In Java, an array can be declared without initialization without mentioning the size. (TRUE / FALSE)**

A) TRUE

B) FALSE

C) -

D) -

Answer [=]

**A**

**Explanation:**

**It is a Lazy initialization of an array.**

**10) What is the output of the below Java code snippet with arrays?**

**static int[] nums;**

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

**{**

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

**}**

A) 0

B) null

C) Compiler error

D) Runtime Exception - Null Pointer Exception

Answer [=]

**D**