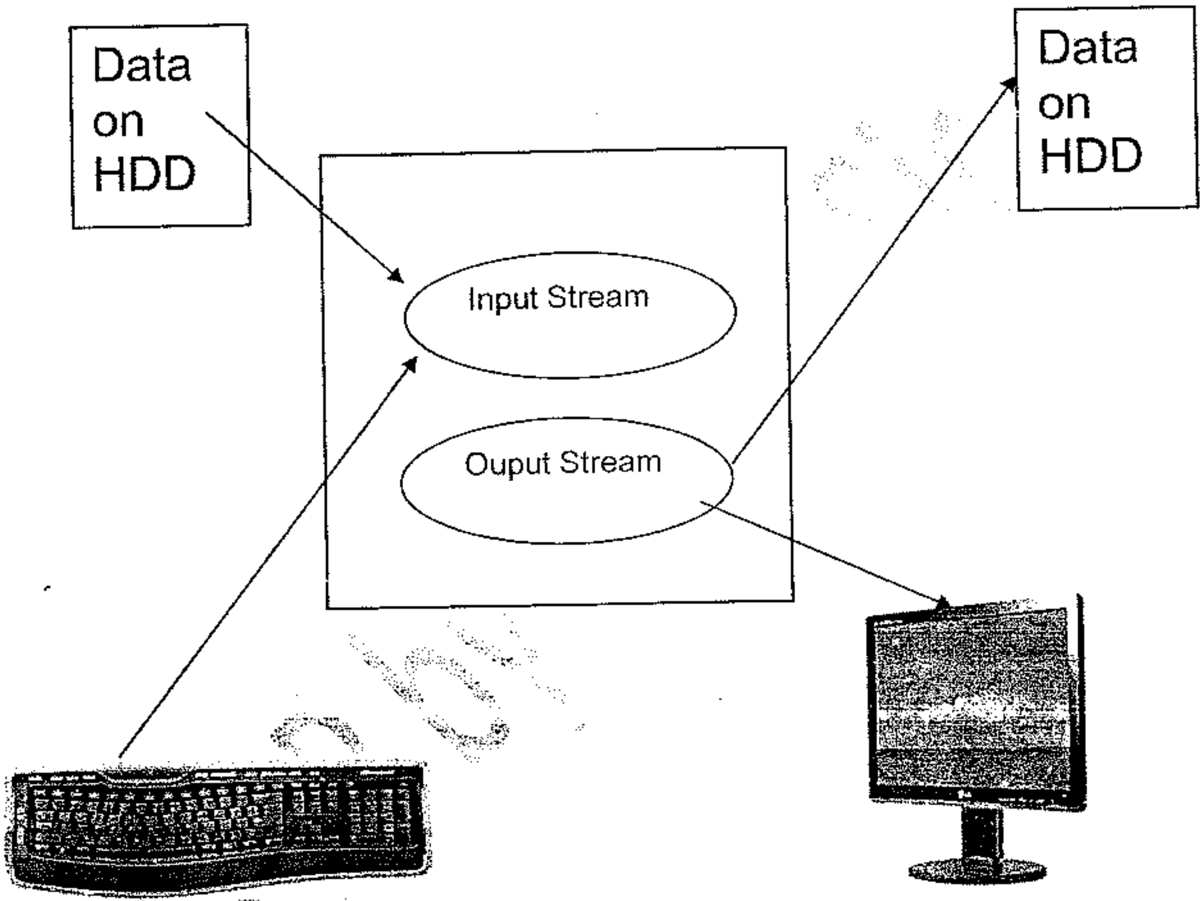
Chapter 5 Java I/O & Arrays

The division of the chapter is as follows:

Chap5 Java I/O and Arrays

Stream:

- A Stream represents a uniform, easy-to-use, object oriented interface between the program and input/output devices
- We can write data to a stream and read data from a stream.



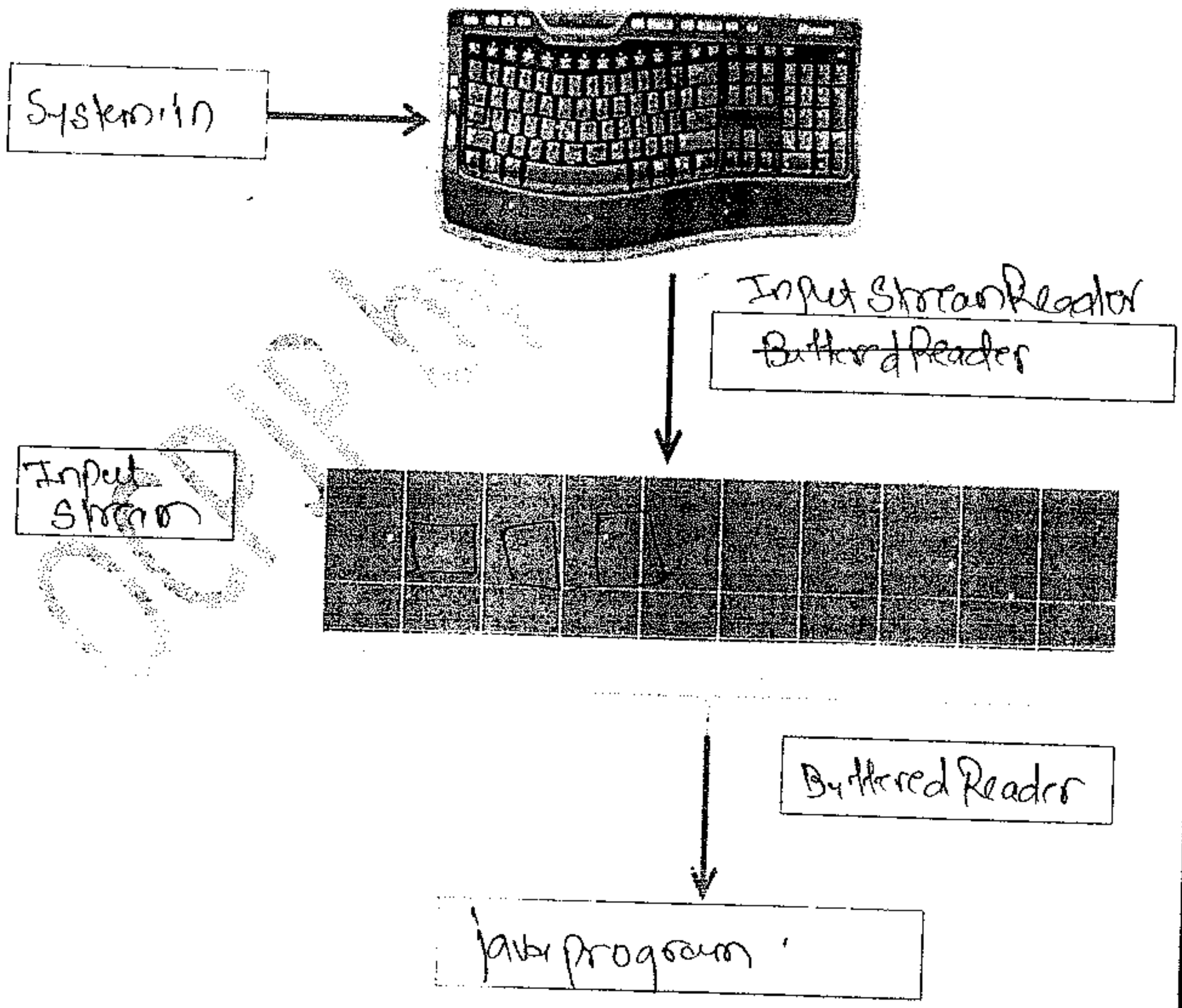
- We read data from **<u>input stream</u>**. The source of input stream can be keyboard, file, etc.
- We write data to **<u>output stream.</u>** The output stream can go to display screen, files etc.
- The use of stream has two advantages:
    - We don't have to worry about the details of each device (would be handled by Java behind the scenes)
    - Program can work for a variety of I/O devices without any changes in the code.
- The package **<u>java.io</u>** contains the classes that provide the foundation for Java's support for stream I/O

InputStreamReader:

- The concrete class that we would use to read an input stream is InputStremReader.
- Eg: InputStreamReader isr = new InputStreamReader(System.in)
  This creates an InputStreamReader object isr from the object System.in, the keyboard input stream.

BufferedReader:

- The operations with a reader can be made more efficient if we buffer it using BufferedReader object.
- Eg: BufferedReader br = new BufferedReader(isr)
  This creates a BufferedReader object br making the input operations more efficient.



Q) WAJP to read an integer from user and print it on the screen.

Program:

```
import java.io.*;
class R1
{
public static void main(String args[]) throws IOException
{
 int a;
 InputStreamReader isr = new InputStreamReader(System.in);
 BufferedReader br = new BufferedReader(isr);
 String s;
  System.out.println("Enter Value");
   s = br.readLine();
  a = Integer.parseInt(s);
  System.out.println("Enter Value="+a);
  }
}
```

O/P:-

```
Javac R1.java
  >java R1
  Enter Value
  5
  Entered Value = 5.
```

Q1) What if we don't write import.java.io.*; ?

Ans: CF ; Because IOException, InputStreamReader and Buffered are defined in that package.

Q2) What if we don't write throws IOException?

Ans: CF, Because IOException must be Caught Or declared.

Q3) What if we don't write System.in?

Ans: CF, ISR does not have a DC.

Q4) What if we don't pass InputStreamReader object to BufferedReader?

Ans: CF, Because BufferedReader does not have a DC.

Q5) What if we don't write use Wrapper classes?

Ans: CF , Because String and Int are incompatible

Q) WAJP to read an integer from user and check whether the number is even or odd.

Program:

```
import java.io.*;
class  EvenOdd
{
        public static void main(String args[]) throws IOException
        {
St 1            int a;
                InputStreamReader isr = new InputStreamReader(System.in);
                BufferedReader br = new BufferedReader(isr);
                String s;
St 2            System.out.println("Enter Value of a ");
                s = br.readLine();
                a = Integer.parseInt(s);
St 3            if(a%2==0)
                {
                S.O.P("Even");
                }
                else
                {
                S.O.P("odd");
                }
        }
  }

        }       // end of main()
} // end of class
```

Output:

```
>javac EvenOdd.java

>java EvenOdd
Enter Value of a
20
Even.
```

Q) WAJP to evaluate an expression : square root of cos(a) + sin(b)
Read the values of a & b from the user.

Program:

```
import java.io.*;
 class  Expr
{
        public static void main(String args[]) throws IOException
        {
St 1          double a=0.0 , b=0.0;
              InputStreamReader isr = new InputStreamReader(System.in);
              BufferedReader br = new BufferedReader(isr);
              String s = new String();

St 2          System.out.println("Enter Value of a ");
              s = br.readLine();
              a = Double.parseDouble(s);

              System.out.println("Enter Value of  b");
              s = br.readLine();
              b =  Double.parseDouble(s);
St 3          double a1 = Math.cos(a);
              double a2 = Math.sin(b)
              double r = Math.sqrt(a1+a2);
              S.o.p ("Result is"+r);
        3}
        }       // end of main()
 } // end of class
```

dynamic initialization bcoz it is initialized at runtime

Output:

```
>javac Expr.java

>java Expr
enter value of a   10
enter value of b   20
Result is 0.4328130100
```

Q) WAJP to read a character to determine whether it is a letter, digit or other character.

Program:

```
import java.io.*;

class Char1 {

        public static void main(String args[ ]) throws IOException  {

                    char ch;

St 1 ——             InputStreamReader isr = new InputStreamReader(System.in);

                    BufferedReader br = new BufferedReader(isr);

                    System.out.println("Enter a character : ");

St 2 ——             ch =(char) br.read(); //Getting ~~book~~ unic Code
                    if (Character.isLetter(ch))
                     S.o.p("the character is a letter");
                    else
                     if (Character.isDigit(ch))
Step3 ——              S.o.p("the charact is a digit");
                      else
                        S.o.p("other Character");
                    }

        } // end of main()
} // end of class
```

Output:

```
>javac Char1.java
>java Char1
Enter a character :
a
the character is a letter
```

```
>java Char1
Enter a character :
3
the character is a digit
```

```
>java Char1
Enter a character :
&
other character
```

```
>java Char1
Enter a character :
1a$
the character is a digit
```

Java Arrays

- Array is group of element of the same type.
- Arrays are objects in Java that store multiple variables of the same type.
- Arrays can hold either primitives or object references, but the array itself will always be an object on the heap, even if the array is declared to hold primitive elements.
- In other words, there is no such thing as a primitive array, but you can make an array of primitives.

Categories of Arrays:

The following are the broad categories of Arrays:

- One Dimensional Array
- Multidimensional Array
    - Two Dimensional Array
    - Three Dimensional Array
    - Four Dimensional Array
    - :
    - :
    - Variable Sized / Jagged array

eclaring 1D Array:

- Arrays are declared by stating the type of element the array will hold (primitive or object) followed by square brackets.
- The square brackets can be to the left or right of the identifier.

Eg:
```
int a[]
~~int []~~ b;
int[]
```

Constructing 1D Array:

- Constructing an array means creating the array object on the heap (where all objects live) ie. doing a new array type.
- To create an array object, Java must know how much space to allocate on the heap, so we must specify the size of the array at creation time.
- The size of the array is the number of elements the array will hold.

Eg:
```
int a[];
a = new int [5];
```

Note: Declaration and Construction step can be combined also.

Eg:
```
int[] b = new int[10];
```

Declaring, Constructing and Initializing 1D Array:

- An array initializer is a shorthand notation for declaring an array and filling it with values, all in a single statement.
- Array initializers are convenient for quickly creating smaller arrays. Instead of using the new keyword, you list the elements of the array in curly braces separated by commas.

Eg:
```
int [] a = {10,30,20};
```

Q1) Can we assign a value to a reference?

Ans: Eg:
```
int a[];
a = 20;
```
CF Incompatible types.

Q2) Can we specify the size of the array along with the reference?

Ans: Eg: `int a[5];` CF ,

Q3) Can we create a negative size Array?

Ans: Eg:
```
int a[ ];
a = new int[-5];
```
P Compiles, java.lang NegativeArrayException,
But JVM will give

Q4) What are default values of array elements?

Ans: The initial Values are the Same as the initial Value Of instance Variables

Q5) Can we initialize array with different elements?

Ans: Eg: `int[ ]  a = {10, 30, 20, "JAVA"};`

CF with error of Incompatible Types.

Q6) While using array Initializers can we specify the size of the array?

Ans: Eg: `int[5]  a = {10, 30, 20};`

CF No we Can not

Q7) Can we reference array by another reference variable of same type?.

Ans: Eg:
```
int[ ]  a = {10, 30, 20};
int b[ ];
b = a;
for(int i : b)
        System.out.print(i + " ");
```
Yes, But there reference type and dimension should be same.

Q8) Can we reference array by another reference variable of another type?

Ans: Eg:
```
int[ ]  a = {10, 30, 20};
float c[ ];
c = a;
```
CF /

No, Incompatible type.

Q9) Can we reference array by another reference variable of another type by type casting ?

Ans: Eg:
```
int[ ] a = {10, 30, 20};
float c[ ];
c = (float)a;
```
inconvertible types.
smaller = float big

Q10) What if we try to access array element which is beyond the array size ?

Ans: Eg:
```
int[ ] a = {10, 30, 20};
a[3] = 40;
```
Prog. compiles, JVM gives Array index out of bound exception.

Q11) What if we try to access array element which is beyond the array size ?

Ans: Eg:
```
int[ ] a = {10, 30, 20};
for (int i = 0; i<=a.length; i++)
System.out.println("element " + i + " " + a[i] );
```
Same as Q-10 Ans.

Pass1D Array to a Method:

```
class Array1Dh {
public static void main(String args[ ]) {
int[ ] a = {10, 30, 20};
sum(a);
}

static void sum(int[ ] ar) {
int total = 0;
for(int i=0; i<ar.length; i++)
      total += ar[i];
System.out.println("Sum of array elements = "  + total );

}}
```

```
Output:
>javac Array1Dh.java
>java Array1Dh
Sum of array elements = 60
```

Another Syntax for passing 1D array to a method:

static void sum (int ...ar)

Q) WAJP to find the largest number of the given n numbers.

Program:

```
import java.io.*;
class LarArra {
public static void main(String[] args) throws IOException  {
int a[],  i, max, n;
InputStreamReader isr = new InputStreamReader(System.in);
87 1  BufferedReader br = new BufferedReader(isr);
String s;

System.out.println("Enter the size of the array");
s = br.readLine();
87 2  n = Integer.parseInt(s);
a = new int[n];

for(i=0; i<n; i++) {          a.length
        System.out.println("Enter element:" + (i+1));
87 3    s = br.readLine();
            a[i] = Integer.parseInt(s);
}

max = a[0];
for(i=1;i<n;i++) {     a.length
        if (a[i] > max)
87 4            max = a[i];
}
System.out.println("Max number is  : "+ max);
}}
```

Output:

```
>javacLarArra.java
> java LarArra
Enter the size of Array 3
Enter element: 1
30
Enter element :2
10
Enter element 3
60
Max Number is: 60
```

Multidimensional Array

2 Dimensional Arrays

- Two dimensional arrays nothing but arrays of arrays.

Declaring 2D Array:

Eg:
```
int[][] a;
int [] b[];
int c[][];
```

Constructing 2D Array:

- A two dimensional array of type int is really an object of type int array (int []) with each element in that array holding a reference to another int array.

Eg:
```
int[][] a;
a = new int[4][5];
```

Declaring, Constructing and Initializing 2D Array

- The following is the way to declaring, constructing and initializing 2D array.

Eg:
```
int a[][] = {{10,20,30,40},{50,60,70,80},
{90,100,110,120}};
```

Q1) While using array Initializers can we specify the size of the array?

Ans: Eg:
```
int a[4][5] = {   {10, 20, 30, 40},
                  {50, 60, 70, 80},
                  {90, 100, 110, 120}
              };
```
No

Q2) Can we reference 2D array by another reference variable of 2D of same type ?

Ans: Eg:
```
int a[ ][ ] = {   {10, 20, 30, 40},
                  {50, 60, 70, 80},
                  {90, 100, 110, 120}
              };
int b[ ][ ];
b = a;
```
Yes

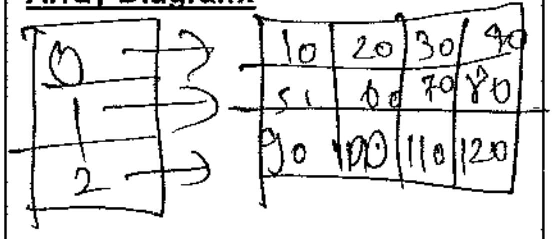
Q3) Can we reference array of 2D by another reference variable of 1D of same type?

Ans: Eg:
```
int a[ ][ ] = {   {10, 20, 30, 40},
                  {50, 60, 70, 80},
                  {90, 100, 110, 120}
              };
int b[ ];
b = a;
```
No
incompatible types.

Passing 2D Array to a Method:

```
class Array2Df {
public static void main(String args[]) {
int a[ ][ ] = {     {10, 20, 30, 40},
                    {50, 60, 70, 80},
                    {90, 100, 110, 120}
          };
sum(a);
}
static void sum(int[ ][ ]  b) {        ith row
int total=0;
for (int i=0; i<b.length; i++) {       column in ith row
          for(int j=0; j<b[i].length; j++) {
                    total += b[i][j];
          }
}
          System.out.println("Total = " + total);
} // end of sum
} // end of class
```

Array Diagram:



Output:

```
>javac Array2Dd.java
>java Array2Dd
Total=780
```

Another Syntax for Passing 2D Array to a Method:

Static void Sum(int[]..., b)

Wrong Syntax for Passing 2D Array to a Method:

static void Sum(int ...[] b)

Variable sized Arrays / Jagged Arrays: [To Save memory]

- A jagged arrays is an array that contains a group of arrays within it.
- It means that we can create an array in Java such that other arrays can become its elements.
- Jagged arrays are also called "irregular dimensional arrays".

Declaring VD Array:

Eg:

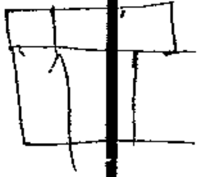
```
int [][]a;
a = new int[4][];
```

Note: specifying only the column will lead to compilation error [i.e int[][] a; a = new int[][4];

Constructing VD Array:

Eg:

```
int[][]a;
a = new int[4][];
a[0] = new int[2];   a[2] = new int[1];
a[1] = new int[3];   a[3] = new int[2];
```



Declaring, Constructing and Initializing VD Array:

Eg:

```
int a[][] = { {10,20},
              {30,40,50,60},
              {70,89,90}
            };
```

Array Diagram:

0 → 10 20

1 → 30 40 50 60

2 → 70 89 90

Arrays Class:

- Arrays class provides methods to perform certain operations on any one dimensional array.
- All the methods of the Arrays class are static, so they can be called in the form Arrays.methodName();
- To use the methods we need to import java.util.*;

Arrays Class Methods:

- **static void sort(array):**
  this methods sorts all the elements of an array into ascending order. This method uses QuickSort Algorithm.

- **static int binarySearch(array, element):**
  this method searches for an element in the array and returns its position number. This method uses BinarySearch Algorithm.

Sorting Array: **Arrays of Strings**

```
import java.util.*;
class MyArraySort  {
        public static void main (String args[]) {
        String[ ] book = {"C", "C++", "Java",
        "Oracle", "Android", "Php"};

        Arrays.sort(book);

        System.out.println("Sorted Array: ");
        for (String s: book)
                System.out.println(s);
        }
}
```

Output:

```
Android
C
c++
Java
oracle
php.
```

Sorting Array: **Arrays of int**

```
import java.util.*;
class MyArraySort1a {
public static void main (String args[]) {
        int[ ] a = {10, 40, 30, 50, 70, 20};

        Arrays.sort(a);

        System.out.println("Sorted Arrays: ");
        for (int i : a)
              System.out.println(i);
        }
}
```

Output:

```
10
20
30
40
50
70
```

Q1) Can we sort the primitives in Descending order?

Ans: No. [Only in ascending]

Q2) Can we sort in String descending order?

Ans: Yes. [Arrays.Sort(book, Collections.reverseOrder());

Q3) Can we sort in Strings in a case insensitive manner?

Ans: Yes. [Arrays.Sort(book, String.CASE_INSENSITIVE_ORDER);

Q4) Can we sort an already sorted array?

Ans: Yes

Q5) Which package should be imported for using Arrays.sort()?

Ans: Java.Util

Q6) Which sorting algorithm does Arrays.sort() use?

Ans: Quick Sort.

Searching Arrays:

```
import java.util.*;
class MyArraySearch
{
public static void main (String args[])
{
String[ ] book = {"C", "C++", "Java", "Oracle",
 "Android", "Php"};

Arrays.sort(book);

System.out.println("Sorted Arrays: ");
for (String s: book)
        System.out.println(s);
System.out.println("Searching Array for Java :  "+
 Arrays.binarySearch(book, "Java") );
        }
}
```

Output:

```
Sorted Arrays:
Android
  C
  C++
  Java
  Oracle
   Php
Searching Array for
   Java : 3
```

Q1) What if the search is unsuccessful?

Ans: it returns a -ve value. ans = (- (insertion point) - 1)

Q2) How do we interpret the value returned?

Ans: ans = (- (insertion point) - 1).

Q3) What if the array that we are searching for is unsorted?

Ans: The result of the search is unpredictable.

Q4) Which package should be imported for using Arrays.binarySearch()?

Ans: Java.Util.

Q5) Which sorting algorithm does Arrays.binarySearch() use?

Ans: binary search.

Test Paper: → IO

Q1)

```
class  R10a
{
public static void main(String[]
args) throws IOException
{
int a;
InputStreamReader isr = new
InputStreamReader(System.in);
BufferedReader br = new
BufferedReader(isr);
String s;
:
:
}
}
```

What will happen when programmer tries to compile and execute the above program?

Options:

[✓ A. Program will not compile.]

B. Program will compile and run successfully.

C. Program will throw IOException

D. JVM will show compilation error.

Solution: A
[import io.*;]

Q2)

```
import java.io.*;

class  R10b {
public static void main(String[]
args)
 {
int a;
InputStreamReader isr = new
InputStreamReader(System.in);
BufferedReader br = new
BufferedReader(isr);
String s;
:
s = br.readLine();
:
}
}
```

What will happen when programmer tries to compile and execute the above program?

Options:

A. Program will compile and run successfully.

[✓ B. Compilation Fails.]

C. Program will throw IOException

D. JVM will show compilation error.

Solution: B [Becoz io:Exception must be ~~Caut~~ Caught or declared.

Q3)

```
1. import java.io.*;
2. class  R10c {
3. public static void main(String[]
args) throws IOException {
4. int a;
5. InputStreamReader isr = new
InputStreamReader();
6. BufferedReader br = new
BufferedReader();
7. String s;
:
:
}
}
```

[illegible]

What will happen when programmer tries to compile and execute the above program?

Options:

A. Program will generate keyboard not found error.

B. Program will show error missing statement.

C. Program will throw IOException

[✓ D. Program will throw error due to constructor not used correctly.]

Solution: D

Q4)

```
1. import java.io.*;
2. class  R10d {
3. public static void main(String[]
args) throws IOException {
4. int a;
5. InputStreamReader isr = new
InputStreamReader(System.in);
6. BufferedReader br = new
BufferedReader(isr);
7. String s;
8. System.out.println("Enter
Value");
9. s = br.readLine();
10. a = s;
11. System.out.println("Entered
value = " + a);
}
}
```

What will happen when programmer tries to compile and execute the above program?

Options:

A. Program will generate keyboard not found error.

[✓ B. Program will show compilation error.]

C. Program will show error missing statement.

D. Program will throw error due to constructor not used correctly.

Solution: B
Incompatible type

Q5)

```
import java.io.*;
class  R10e
{
public static void main(String[] args)
throws IOException
{
int a;
InputStreamReader isr = new
InputStreamReader(System.in);
BufferedReader br = new
BufferedReader(isr);
String s;
System.out.println("Enter Value");
s = br.readLine();
a = Integer.parseInt(s);
System.out.println("Entered value = "
+ a);
}
}
```

What will happen when programmer tries to compile and execute the above program?

Options:

A. Program will compile and run successfully.

B. Program will show compilation error.

C. JVM will report compilation error.

D. Program will throw error due to constructor not used correctly.

Solution: A

Q6)

```
class  Expr10a {
public static void main(String[] args) {
char c1 = 'a', c2 = '5', c3 = '&';
System.out.print(Character.isLetter(c1));
System.out.print(Character.isDigit(c2));
System.out.print(Character.isLetter(c3));
}}
```

Options:

A.truetruetrue

B. falsefalsefalse

C. truetruefalse

D. truefalsefalse

Solution: C

Test Paper: ➔ Arrays

Q1) Which of the following statements are valid array declaration?

(A) int number( ); []
(B) float average[ ];
(C) double[] marks;
(D) counter int[ ];

Options:

A. (A)
[✓ B. (B) & (C)]
C. (A) & (C)
D. (D)

Solution: B

Q2)
Which three are legal array declarations?

1. int [ ] myScores [ ];
2. char [ ] myChars;
3. int [6] myScores;
4. Dog myDogs [ ];
5. Dog myDogs [7];

Options:

[✓ A. 1, 2, 4]
B. 2, 4, 5
C. 2, 3, 4
D. All are correct.

Q3) Which will legally declare, construct, and initialize an array?

Options:

A. int [ ] myList = {"1", "2", "3"};
B. int [ ] myList = (5, 8, 2);
C. int myList [ ][ ] = {4,9,7,0};
[✓ D. int myList [ ] = {4, 3, 7};]

Solution: D

Q4) Which of the following(s) will cause a compiler error?
Select one correct answer.

Options:

A. int[ ] scores = {3, 5, 7};

B. int [ ][ ] scores = {2,7,6}, {9,3,45};

C. String cats[ ] = {"Fluffy", "Spot", "Zeus"};

D. boolean results[ ] = new boolean [ ] {true, false, true};

Solution: B

Q5) Consider the following code

```
int number[ ] = new int[5];
```

| 0 | 0 | 0 | 0 | 0 |

After execution of this statement, which of the following are true?

(A) number[0] is undefined
(B) number[4] is 0
(C) number[4] is null
(D) number[2] is 0
(E) number.length is 5

Options:

A. (A) & (E)
B. (B), (D) & (E)
C. (C) & (E)
D. (E)

Solution: B

Q6) Which one of the following array declaration statements is not legal?

Options:

(a) int [ ]a[ ] = new int [4][4];
(b) int a[ ][ ] = new int [4][4];
(c) int a[ ][ ] = new int [ ][4];
(d) int [ ]a[ ] = new int [4][];
(e) int [ ][ ]a = new int [4][4];

Solution: C

Q7) Suppose we declare the integer array, marks for handling the marks of the students. We use the following code to declare and initialize the marks array:

```
int[ ] marks;
marks = new int[10];
```

Which of the following option will get the size of the marks array?

Options:

A. marks[ ].length
B. marks.length()
C. marks.length
D. marks.size()

Solution: C

Q8) What will be the output of the program?

```
public class ArrayTest {
   public static void main(String[ ] args) {
      float f1[ ], f2[ ];
      f1 = new float[10];
      f2 = f1;
      System.out.println("f2[0] = " + f2[0]);
   } }
```

Options:

A. It prints f2[0] = 0.0
B. It prints f2[0] = NaN
C. An error at f2 = f1; causes compile to fail.
D. It prints the garbage value.

Solution: A

Q9) Given:

```
1. class Alligator {
2. public static void main(String[ ] args) {
3. int [ ]x[ ] = {{1,2}, {3,4,5}, {6,7,8,9}};
4. int [ ][ ]y = x;
5. System.out.println(y[2][1]);
6. }    7. }
```

0 0 1 1 2 3

1 3 2 4 5

2 6 7 8 9

What is the result?

Options:

A. 2 B. 3 C. 4

D. 6 E. 7 F. Compilation fails.

Solution: E

Q10)

```
public class F0091 {
   public void main( String[ ] args ) {
      System.out.println( "Hello" + args[0] );
}}
```

What will be the output of the program, if this code is executed with the command line: **> java F0091 world**

Options:

A. Hello

B. Hello Foo91

C. Hello world

[✓ D. The code does not run.]

No static

Solution:

Q11) What will be the output of the program?

```
public class CommandArgsTwo {
   public static void main(String [ ] argh)  {
      int x;
      x = argh.length;
      for (int y = 1; y <= x; y++) {
         System.out.print(" " + argh[y]);
      }
   }
}
```

and the command-line invocation is

> java CommandArgsTwo 1 2 3

O/P 2 3 AIOB

Options:

A. 0 1 2

B. 1 2 3

C. 2 3

[✓ D. An exception is thrown at runtime]

Solution: D

Q12) Given:

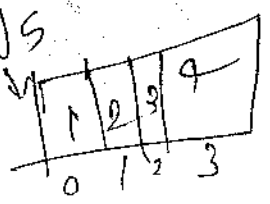
```
public class Yippee {
public static void main(String [ ] args) {
for(int x = 1; x < args.length; x++) {
System.out.print(args[x] + " ");
}
}
}
```

and two separate command line invocations:

java Yippee
java Yippee 1 2 3 4

What is the result?

① no o/p
② args



Options:
A. 1 ~~2 3~~
[✓ B. 2 3 4]
C. 1 2 3 4
D. 1 2 3 An exception is thrown at runtime.
E. 2 3 4 An exception is thrown at runtime.
F. 1 2 3 4 An exception is thrown at runtime.

Solution:

Q13) Given this code in a method:

```
4. int x = 0;
5. int[ ] primes = {1,2};
6. for(int i: primes)
7. switch(i) {
8. case 1: x += i;
9. case 5: x += i;
10. default: x += i;
11. case 2: x += i;
12. }
13. System.out.println(x);
```

x=0
i=1
0+1=1
1+1=2
2+1=3
3+1=4

i=2
x=4+2
=6

What is the result?

Options:
A. 11
~~B. 13~~
C. 24
[✓ D. 6]
E. Compilation fails due to an error on line 7.
F. Compilation fails due to an error on line 10.
G. Compilation fails due to an error on line 11.

Solution:

Q14)

```
public class Venus1 {
public static void main(String[ ] args) {
int [ ] x = {1,2,3};
int y[ ] = {4,5,6};
new Venus1().go(x);
}
void go(int... z) {
for(int i : z)
System.out.print(z[0]); // 7' then o/p = 123
}
}
```

What is the result?

Options:

A. 1 [✓ B. 111] C. 3
D. 123 E. Compilation fails. F. An exception is thrown at runtime.

Solution:

Q15)

```
import java.util.*;
public class Test1 {
public static void main(String args[ ]) {
String[] sa = {"foo", "bat", "ball"};
// insert code
}}
```

Which statement from the following options can be replaced with comment provided in the precedeing program to sort the a array?

Options:

[✓ A. Arrays.sort(sa)] B. sa.sort()
C. Arrays.sort() D. Arrays.Sort(sa)

Solution:

Q16) What is the result of the following statements?

```
10. int [ ] random = {6, -4, 12, 0, -10};
11. int x = 12;
12. int y = Arrays.binarySearch(random, x);
13. System.out.println(y);
```

Options:

A. 2 B. 4
C. The result is undefined. D. Line 12 throws an exception at runtime.
E. Compiler error on line 12

[illegible] fails

Solution: C

C

Q17) Given:

```
import java.util.*;
public class Quest {
public static void main(String[ ] args) {
String[ ] colors = {"blue", "red", "green", "yellow", "orange"};
Arrays.sort(colors);
int s2 = Arrays.binarySearch(colors, "orange");
int s3 = Arrays.binarySearch(colors, "violet");
System.out.println(s2 + " " + s3);
}
}
```

What is the result?

0 b
1 g 2
2 o
3 r V
4 y

Options:

A. 2 -1 B. 2 -4 C. 2 -5
D. 3 -1 E. 3 -4 F. 3 -5
G. Compilation fails. H. An exception is thrown at runtime.

Solution: C

Q18) Given:

```
public class Q25{
 static int[ ] a;
 public static void main( String[ ] args ) {
 a = new int[-5];
 }}
```

Which exception or error will be thrown when a programmer attempts to run this code?

Options:

A. java.lang.NullPointerException B. java.lang.NegativeArraySizeException
C. java.lang.ExceptionInInitializerError D. java.lang.ArrayIndexOutOfBoundsException

Solution: B

Q19) Given:

```
public class Q117d {
static int[ ] a;
public static void main( String[ ] args ) {
a = new int[5];
a[0] = 2;
a[2] = 3;
a[-1] = 2;
} }
```

Which exception or error will be thrown when a programmer attempts to run this code?

Options:

A. java.lang.NullPointerException B. java.lang.NegativeArraySizeException
C. java.lang.ExceptionInInitializerError D. java.lang.ArrayIndexOutOfBoundsException

Solution: D

Q20) Given:

```
10. public class Q117c {
11. static int[ ] a;
12. public static void main( String[ ] args ) {
13. a[-1] = 2;
14. }   15. }
```

Which exception or error will be thrown when a programmer attempts to run this code?

Options:

[✓ A. java.lang.NullPointerException] B. java.lang.NegativeArraySizeException
C. java.lang.ExceptionInInitializerError D. java.lang.ArrayIndexOutOfBoundsException

If this code in static block then it will appears [error in static block]

Solution:

Q21)

```
11. class Mud {
12. // insert code here
13. System.out.println("hi");
14. }   15. }
```

And the following five fragments:

✓ public static void main(String ...a) {
public static void main(String.* a) {
✓ public static void main(String... a) {
✓ public static void main(String[]... a) {
public static void main(String...[] a) {

How many of the code fragments, inserted independently at line 12, **compile**?

Options:

A. 0 B. 1 C. 2
[✓ D. 3] E. 4

Solution:

Q22)

```
11. class Mud {
12. // insert code here
13. System.out.println("hi");
14. }15. }
```

And the following five fragments:

✗ public static void main(String.* a) { ✗
✓ public static void main(String ...a) { ✓
✓ public static void main(String... a) { ✓
✗ public static void main(String...[] a) { ✗
✓ public static void main(String[]... a) { ✗

[JVM needs array of String of 1 dimension

How many of the code fragments, inserted independently at line 12, **compile and run**?

Options:

A. 0 B. 1 [✓ C. 2]
D. 3 E. 4

Solution: