

## JAVA NOTES:

### BASICS:

The screenshot shows a Java code editor window titled "Test.java". The code demonstrates reference type assignment:

```
18     y.cost=1300000;
19     System.out.println(y.cost);
20     System.out.println(y.name);
21     System.out.println(x.name);
22     System.out.println(x.cost);
23
24
25
26
27 }
28
29
30 }
```

The code creates two objects, x and y, of type Car. It then assigns the value of x to y using the assignment operator (=). When printed, both y and x show the same name ("Civic") and cost (1300000).

The screenshot shows a Java code editor window titled "Test.java" and a "Console" window. The code is identical to the one above, demonstrating reference type assignment:

```
1 =/*REFERENCE TYPE ASSINGMENT
2 Address Present in one variable will also affect the another variable*/
3
4 class Car {
5     String name;
6     int cost;
7 }
8 public class Test{
9     public static void main(String[] args) {
10         Car x=new Car();
11         x.name="Civic";
12         x.cost=1200000;
13         System.out.println(x.name);
14         System.out.println(x.cost);
15         Car y;
16         y=x;
17         y.name="City";
18         y.cost=1300000;
19         System.out.println(y.cost);
20         System.out.println(y.name);
21         System.out.println(x.name);
22         System.out.println(x.cost);
23
24
25
26
27 }
28
29
30 }
```

The "Console" window shows the output of the program:

```
Civic
1200000
1300000
City
City
1300000
```

## JAVA NOTES:

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Ulooping.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

StringPalin... MethodOver... Printer.java Encapsulat... Calculator.java BankAccount... Customer.java Constructor... MaxJava CarJava Alerts

Console

```
1 public class Ulooping {  
2     public static void main(String[] args) {  
3         System.out.println("U"+ String.format("%87d",353));  
4     }  
5 }  
6  
7 }
```

terminated: Ulooping [Java Application] D:\Program Files (x86)\IBM\SDP\jdk\bin\javaw.exe (Nov 3, 2023, 2:24:30 PM)  
10000353

Type here to search Share this window

2:25 PM 11/3/2023 1425 ENG IN 03-11-2023

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/ValueTypeAssignment.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Proj Run Window Help

StringPalin... MethodOver... Printer.java Encapsulat... Calculator.java BankAccount... Customer.java Constructor... MaxJava CarJava Alerts

Console

```
1 /* Value Type Assignment  
2 the value present in one variable does not affect the another variable.*/  
3  
4 public class ValueTypeAssignment {  
5     public static void main(String[] args) {  
6         int x=100;  
7         int y;  
8         y=x;  
9         System.out.println(x);  
10        System.out.println(y);  
11        x=200;  
12        System.out.println(x);  
13        System.out.println(y);  
14    }  
15 }  
16  
17  
18  
19 }
```

terminated: ValueTypeAssignment [Java Application] D:\Program Files (x86)\IBM\SDP\jdk\bin\javaw.exe (Nov 3, 2023, 2:25:30 PM)  
100  
100  
200  
100

Type here to search Share this window

2:26 PM 11/3/2023 1425 ENG IN 03-11-2023

## JAVA NOTES:

### INSTANCE VARIABLE:

The screenshot shows the Rational Application Developer for WebSphere interface. The left pane displays the Enterprise Explorer with a project named 'AMAR' containing a 'src' folder with files like 'Dog.java', 'LocalVariable.java', and 'String.java'. The right pane shows a code editor with the following Java code:

```
another variable.*/
18
19 }
20
```

The status bar at the bottom indicates '0 items selected' and shows system information: '2:29 PM 11/11/2023 ENG IN 14:20 03-11-2023'.

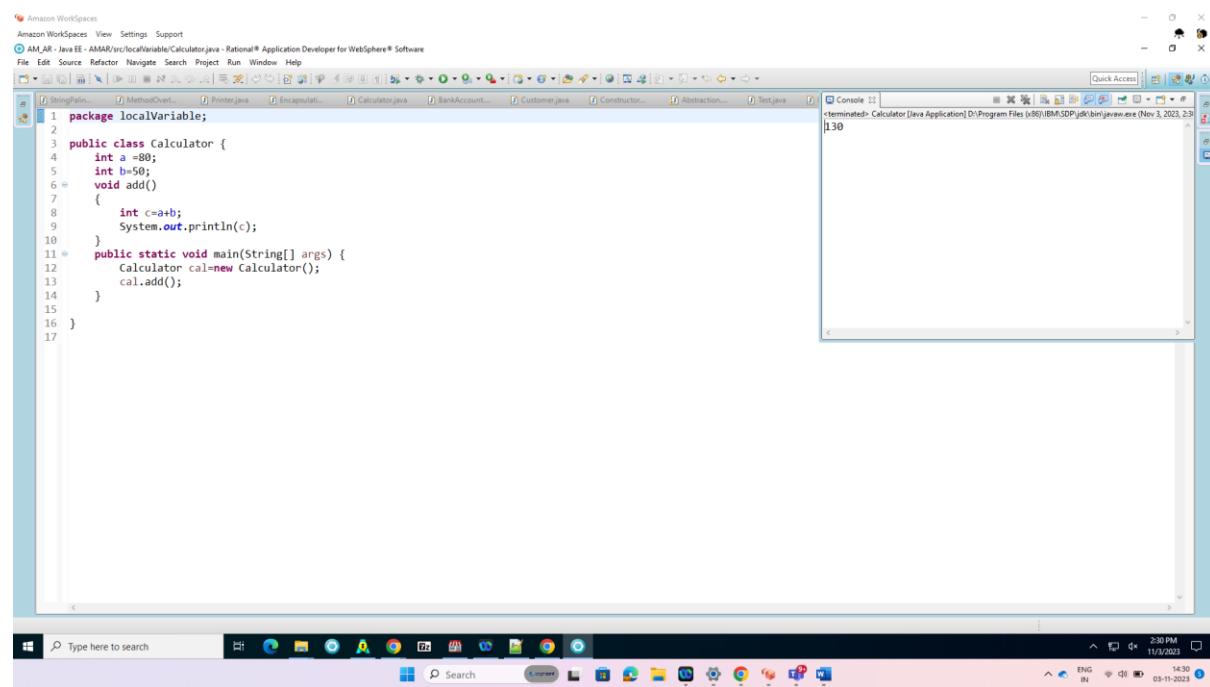
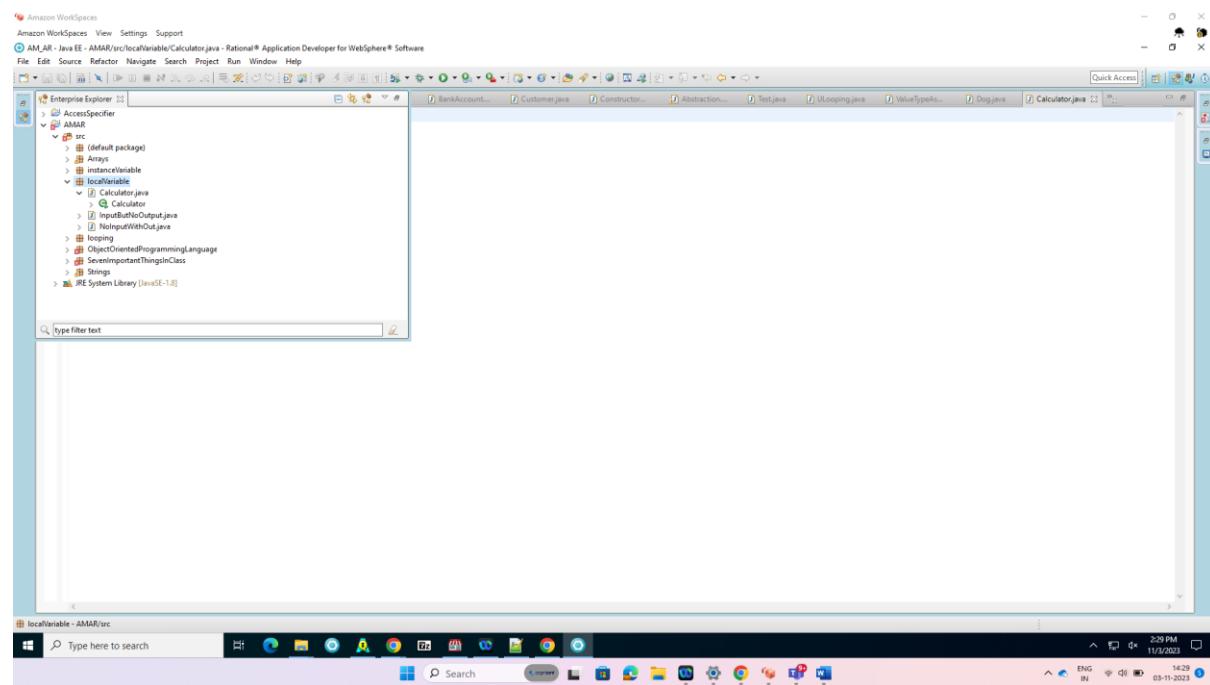
The screenshot shows the Rational Application Developer for WebSphere interface. The left pane displays the Enterprise Explorer with a project named 'AMAR' containing a 'src' folder with files like 'Dog.java', 'LocalVariable.java', and 'String.java'. The right pane shows a code editor with the following Java code:

```
1 package instanceVariable;
2 // NOTE THIS IS A LOCAL VARIABLE NOT INSTANCE
3
4 public class Dog {
5
6     int bearableCost = 3000;
7     String name ="tummy";
8     String person ="human";
9     public static void main(String[] args) {
10         Dog d=new Dog();
11         System.out.println(d.name) ;
12     }
13 }
14
15 }
```

Below the code editor is a 'Console' window showing the output of a run: 'terminated> Dog [Java Application] D:\Program Files (x86)\IBM\SDP\jdk\bin\java.exe (Nov 3, 2023, 2:29:06 PM)' followed by the text 'tummy'. The status bar at the bottom indicates '2:29 PM 11/11/2023 ENG IN 14:20 03-11-2023'.

## JAVA NOTES:

### LOCAL VARIABLE:



## JAVA NOTES:

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AMAR - Java EE - AMAR/src/localVariable/InputButNoOutput.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

StringP... MethodOver... Printer.java Encapsulat... Calculator.java BankAccount... Customer.java Abstraction... TestJava ULooping.java

Console

<terminated> InputButNoOutput [Java Application] D:\Program Files (x86)\IBM\SDP\jdk\bin\java.exe (New 3, 60)

```
1 package localVariable;
2
3 public class InputButNoOutput {
4     int c;
5     void add(int a,int b)
6     {
7         c=a+b;
8         System.out.println(c);
9     }
10    public static void main(String[] args) {
11        InputButNoOutput calc=new InputButNoOutput();
12        int num1 = 20;
13        calc.add(num1, 40);
14    }
15
16
17 }
18
```

Type here to search

2:31 PM 11/13/2023

ENG IN 14:31 03-11-2023

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AMAR - Java EE - AMAR/src/localVariable/NoInputWithOut.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

StringP... MethodOver... Printer.java Encapsulat... Calculator.java BankAccount... Customer.java Abstraction... TestJava ULooping.java ValueTypesAs...

Console

<terminated> NoInputWithOut [Java Application] D:\Program Files (x86)\IBM\SDP\jdk\bin\java.exe (New 3, 20)

```
130
```

```
1 package localVariable;
2
3 public class NoInputWithOut {
4     int a =50;
5     int b=80;
6     int add()
7     {
8         int c=a+b;
9         return c;
10    }
11    public static void main(String[] args) {
12        NoInputWithOut cal=new NoInputWithOut();
13        cal.add();
14        System.out.println(cal.add());
15    }
16
17 }
```

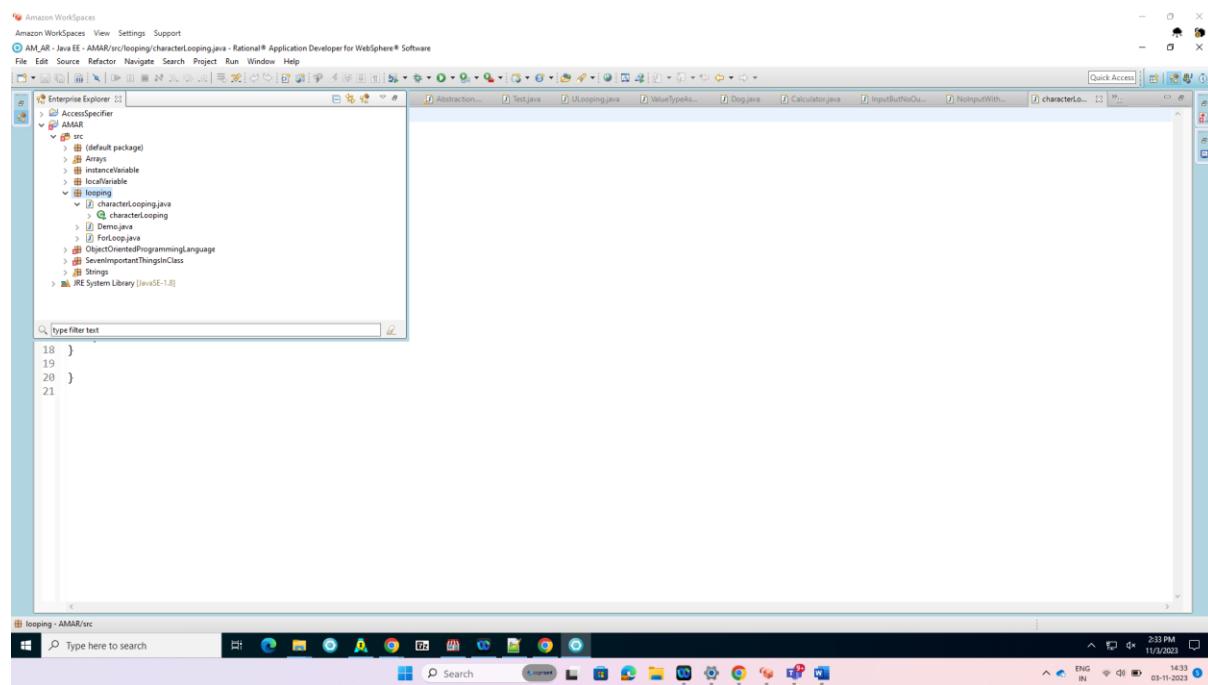
Type here to search

2:32 PM 11/13/2023

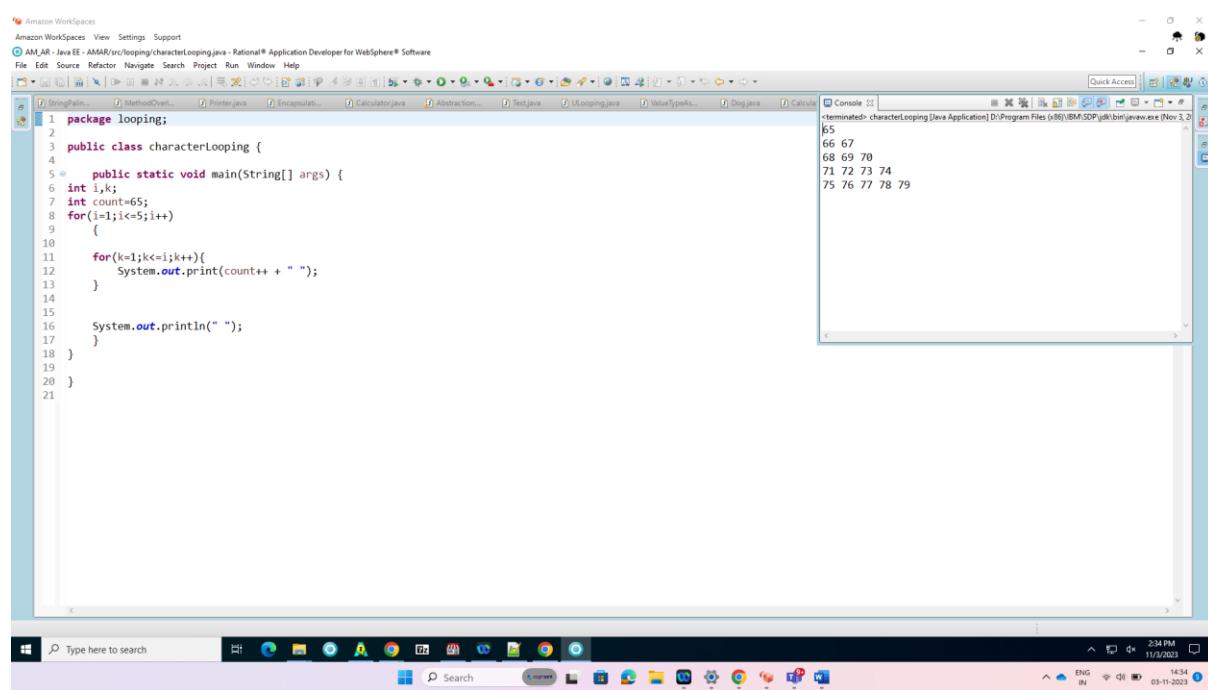
ENG IN 14:32 03-11-2023

## JAVA NOTES:

### LOOPING:



```
18 }
19 }
20 }
21 }
```



```
1 package looping;
2
3 public class characterLooping {
4
5     public static void main(String[] args) {
6         int i,k;
7         int count=65;
8         for(i=1;i<=5;i++)
9         {
10             for(k=1;k<=i;k++){
11                 System.out.print(count++ + " ");
12             }
13
14
15             System.out.println(" ");
16         }
17     }
18 }
19
20 }
```

```
65
66 67
68 69 70
71 72 73 74
75 76 77 78 79
```

## JAVA NOTES:

The screenshot shows a Java application running in Rational Application Developer for WebSphere. The code in the editor is a while loop that prints numbers from 1 to 5. The output window shows the results of the execution.

```
5 * * @author AMARNATH.GORLA
6 package looping;
7 public class Demo{
8     public static void main(String[] args) {
9         int i,j;
10        for (i=1;i<=5;i++)
11        {
12            if (i%2==0)
13            {
14                System.out.print(i+1);
15            }
16            for(j=1;j<=5;j++)
17            {
18                System.out.print(i);
19            }
20            if(i%2!=0)
21            {
22                System.out.print(i+1);
23            }
24            System.out.println("");
25        }
26    }
27
28
29
30
31    System.out.println("");
32
33}
34
35
36
37
38
```

Console Output:

```
111112
322222
333334
544444
555556
```

The screenshot shows a Java application running in Rational Application Developer for WebSphere. The code in the editor is a for loop that prints numbers from 10 to 34. The output window shows the results of the execution.

```
1 package looping;
2
3 public class Forloop {
4     public static void main(String[] args) {
5         int i,j,count = 10;
6         for (i=1;i<=5;i++)
7         {
8             for (j=1;j<=5;j++)
9                 if(count<10)
10                 {
11                     System.out.println("0");
12                 }
13                 else
14                 {
15                     System.out.print(count++ + " ");
16                 }
17
18         }
19
20         System.out.println("");
21     }
22
23
24
25
26
27}
```

Console Output:

```
10 11 12 13 14
15 16 17 18 19
20 21 22 23 24
25 26 27 28 29
30 31 32 33 34
```

## JAVA NOTES:

### ARRAY:

The screenshot shows the Rational Application Developer interface. The left pane displays the 'Enterprise Explorer' with a project named 'AMAR' containing a 'src' folder with various Java files like D1.java, D2.java, D3.java, Demo.java, etc. The right pane shows the code for 'ForLoop.java':

```
18     }
19
20     System.out.println("");
21 }
22
23
24
25 }
26
27 }
```

The status bar at the bottom indicates the date as 11/3/2013 and the time as 14:38.

The screenshot shows the Rational Application Developer interface. The left pane displays the 'Enterprise Explorer' with a project named 'AMAR' containing a 'src' folder with various Java files like Demo.java, BankAccount.java, Customer.java, etc. The right pane shows the code for 'Demo.java':

```
1 /*METHODS PRESENT INSIDE THE SCANNER CLASS
2 * ADDITION
3 * SUBTRACTION
4 * MULTIPLICATION
5 * DIVISION
6 * USING ARRAY
7 */
8
9
10 package Arrays;
11
12 import java.util.Scanner;
13
14 public class Demo {
15     public static void main(String[] args) {
16         Scanner scan = new Scanner(System.in);
17         System.out.println("enter num 1:");
18         int a = scan.nextInt();
19         System.out.println("enter num 2:");
20         int b = scan.nextInt();
21         int c=a+b;
22         System.out.println("The Sum Of Number is:" + c );
23     }
24 }
25
26
27
28 /*public class Demo{
29     public static void main(String[] args) {
30         Scanner scan=new Scanner(System.in);
31         System.out.println("Enter the Number:");
32         int a=scan.nextInt();
33         System.out.println("enter the number :");
34         int b = scan.nextInt();
35         int c=a+b;
36         System.out.println("THE MULTIPLICATION OF THE NUM A AND B IS :" +c);
```

The 'Console' tab on the right shows the output of the program:

```
enter num 1:
12
enter num 2:
10
The Sum Of Number is:22
```

The status bar at the bottom indicates the date as 11/3/2013 and the time as 14:38.

## JAVA NOTES:

A screenshot of the Rational Application Developer for WebSphere interface. The top menu bar includes 'Amazon WorkSpaces', 'File', 'Edit', 'Source', 'Refactor', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. The toolbar has icons for file operations like Open, Save, and Print. The main workspace shows a Java code editor with the following code:

```
1 package Arrays;
2
3 import java.util.Scanner;
4
5 public class D1 {
6
7     public static void main(String[] args) {
8         int[] a=new int[5];
9         a[1]=24;
10        System.out.println(a[1]);
11    }
12 }
13
14 // I want to store age of 5 students
15 /*public static void main(String[] args) {
16     int[] a=new int[5];
17
18     Scanner s= new Scanner(System.in);
19
20     System.out.println("enter the age:");
21     a[0]=s.nextInt();
22     System.out.println("enter the age :");
23     a[1]=s.nextInt();
24     System.out.println("enter the age:");
25     a[2]=s.nextInt();
26     System.out.println("enter the age:");
27     a[3]=s.nextInt();
28     System.out.println("enter the age:");
29     a[4]=s.nextInt();
30     System.out.println("enter the age:");
31     a[5]=s.nextInt();
32     System.out.println("enter the age:");
33     a[6]=s.nextInt();
34     System.out.println("enter the age:");
35 }
```

The right side of the interface features a 'Console' window with the output:

```
terminated> D1 [Java Application] D:\Program Files (x86)\IBM\SDP\jdk\bin\java.exe (Nov 3, 2023, 6:55:07 PM)
24
```

The taskbar at the bottom shows various application icons, and the system tray indicates the date as 11/11/2023 and the time as 6:55 PM.

A screenshot of the Rational Application Developer for WebSphere interface, similar to the previous one but with an error. The code is identical to the first screenshot. The 'Console' window now displays an exception:

```
terminated> D1 [Java Application] D:\Program Files (x86)\IBM\SDP\jdk\bin\java.exe (Nov 3, 2023, 6:56:20 PM)
enter the age:
12
enter the age :
12
enter the age:
13
enter the age:
4
enter the age:
34
enter the age:
|
45
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException
at Arrays.D1.main(D1.java:31)
```

The taskbar at the bottom shows various application icons, and the system tray indicates the date as 11/11/2023 and the time as 6:57 PM.

## JAVA NOTES:

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Array/D1.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

StringPalin... MethodOver... Printer.java Encapsulation... Calculator.java BankAccount... Customer.java Constructor... Max.java Car.java Snip...

Console

```
the STUDENTS AGE ARE:0
1
THE STUDENTS AGE ARE:1
2
THE STUDENTS AGE ARE:2
3
THE STUDENTS AGE ARE:3
4
THE STUDENTS AGE ARE:4
5
THE STUDENTS AGE ARE:5
6
THE STUDENTS AGE ARE:6
7
THE STUDENTS AGE ARE:7
8
THE STUDENTS AGE ARE:8
9
THE STUDENTS AGE ARE:9
10
THE AGE IS:1
THE AGE IS:2
THE AGE IS:3
THE AGE IS:4
THE AGE IS:5
THE AGE IS:6
THE AGE IS:7
THE AGE IS:8
THE AGE IS:9
THE AGE IS:10
```

Type here to search

6:59 PM 11/6/2023 18:59 03-11-2023

This screenshot shows a Java application running in Rational Application Developer. The code reads ten student ages from the user and prints them out. The console output shows the ages from 0 to 10.

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Array/D1looping.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

StringPalin... MethodOver... Printer.java Encapsulation... Calculator.java BankAccount... Customer.java Constructor... Max.java Car.java Snip...

Console

```
/*LOOPING THE ARRAYS TO ACCESS IT EASILY AND TO REDUCE THE LINE OF CODES*/
1
2
3 package Arrays;
4
5 import java.util.Scanner;
6
7 public class D1looping {
8     public static void main(String[] args) {
9         int []a= new int[4];
10
11         Scanner s=new Scanner(System.in);
12         for (int i=0;i<a.length-1;i++)
13         {
14             System.out.println("Enter the age: ");
15             a[i]=s.nextInt();
16         }
17
18         System.out.println("The ages are :");
19
20         for(int i=0;i<a.length;i++)
21         {
22             System.out.println(a[i]);
23         }
24     }
25 }
```

Enter the age:
1
Enter the age:
2
Enter the age:
3
Enter the age:
4
Enter the age:
5
The ages are :
1
2
4
5

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException
at Arrays.D1looping.main(D1looping.java:22)

Type here to search

7:00 PM 11/6/2023 19:00 03-11-2023

This screenshot shows a Java application running in Rational Application Developer. The code uses a for loop to read four student ages from the user and print them out. An exception is thrown at line 22 due to an array index out of bounds error.

## JAVA NOTES:

Amazon WorkSpaces View Settings Support  
AMAR - Java EE - AMAR/src/Arrays/D2.java - Rational® Application Developer for WebSphere® Software  
File Edit Source Refactor Navigate Search Project Run Window Help

```
1 /*WRITE A PROGRAM TO STORE AGE OF STUDENTS, 2 CLASSROOMS AND
2
3 TWO DIMENSIONAL ARRAY
4 1D- a.length -> acts as column
5 2D- a[i].length -> acts as row and column.
6
7 EACH CLASS HAVE 5 STUDENTS*/
8
9 package Arrays;
10
11 import java.util.Scanner;
12
13 public class D2 {
14     public static void main(String[] args) {
15         int[][] a=new int[2][5];
16         Scanner scan=new Scanner(System.in);
17         for(int i=0;i<a.length-1;i++)
18         {
19             for(int j=0;j<a[i].length-1;j++)
20             {
21                 System.out.println("The age of the students in classroom " + i + " and Students " + j);
22                 a[i][j]=scan.nextInt();
23             }
24         }
25         System.out.println("THE AGES ARE:");
26         for (int i=0;i<a.length-1;i++)
27         {
28             for(int j=0;j<a[i].length-1;j++)
29             {
30                 System.out.println(a[i][j]);
31             }
32         }
33     }
34
35
36 }
```

Console Output:  
The age of the students in classroom 0 and Students 0  
1  
The age of the students in classroom 0 and Students 1  
2  
The age of the students in classroom 0 and Students 2  
3  
The age of the students in classroom 0 and Students 3  
4  
The age of the students in classroom 0 and Students 4  
5  
The age of the students in classroom 1 and Students 0  
6  
The age of the students in classroom 1 and Students 1  
7  
The age of the students in classroom 1 and Students 2  
8  
The age of the students in classroom 1 and Students 3  
9  
The age of the students in classroom 1 and Students 4  
10  
The age of the students in classroom 1 and Students 5  
11  
THE AGES ARE:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37

Type here to search      7:50 PM 11/3/2023      ENG IN 19:10 03-11-2023

Amazon WorkSpaces View Settings Support  
AMAR - Java EE - AMAR/src/Arrays/D3.java - Rational® Application Developer for WebSphere® Software  
File Edit Source Refactor Navigate Search Project Run Window Help

```
1 /* 3 D ARRAY */
2
3 package Arrays;
4
5 import java.util.Scanner;
6
7 public class D3 {
8     public static void main(String[] args) {
9         int[] [] [] a=new int[2][2][3];
10        Scanner scan=new Scanner(System.in);
11        for(int i=0;i<a.length-1;i++)
12        {
13            for(int j=0;j<a[i].length-1;j++)
14            {
15                for(int k=0;k<a[i][j].length-1;k++)
16                {
17                    System.out.println("THE STUDENT IN SCHOOL " + (i+12) + " IN THE CLASS " + j+1 + " AND THE STUDENT NUMBER " + k+1);
18                    a[i][j][k]=scan.nextInt();
19                }
20            }
21        }
22    }
23
24
25 }
```

Console Output:  
THE STUDENT IN SCHOOL 12 IN THE CLASS 0:  
1  
THE STUDENT IN SCHOOL 12 IN THE CLASS 0:  
2  
THE STUDENT IN SCHOOL 12 IN THE CLASS 0:  
3  
THE STUDENT IN SCHOOL 13 IN THE CLASS 0:  
2  
THE STUDENT IN SCHOOL 13 IN THE CLASS 0:  
3  
THE STUDENT IN SCHOOL 13 IN THE CLASS 0:  
4  
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3  
at Arrays.D3.main(D3.java:14)

Type here to search      7:52 PM 11/3/2023      ENG IN 19:12 03-11-2023

## JAVA NOTES:

A screenshot of the Rational Application Developer interface. The main window shows a Java code editor with the following code:

```
1 package Arrays;
2
3 import java.util.Scanner;
4
5 // Array store only homogenous type data (same type data)
6 // Array is fixed in nature (int[] a=new int[5]; -> a[5] =17 -> it wont print the value coz it is out of range.
7
8 public class DisAdvantagesOfArray {
9     public static void main(String[] args) {
10         Scanner s=new Scanner(System.in);
11         float []a=new float[6];
12         a[1]=2;
13         a[3]=14;
14         a[2]=12;
15         a[4]= 9.81f; /*ARRAY STORE ONLY SAME TYPE OF DATA*/
16         System.out.println(a[2]);
17     }
18 }
```

To the right of the code editor is a terminal window titled "Console" showing the output of the program: "12.0". Below the editor is a standard Windows taskbar with various icons.

A screenshot of the Rational Application Developer interface, showing a different project structure in the Enterprise Explorer:

- AMAR
- src
  - (default package)
  - Arrays
    - D1.java
    - D1looping.java
    - D2.java
    - D3.java
    - Demo.java
    - DisAdvantagesOfArray.java
    - HackthonQuestions.java
    - JaggedArray.java
  - CollectionOfFrameWorks
  - DAO
  - ExceptionsHandling
  - FileHandling
  - instanceVariable
  - IDBC
  - JSCLASS

The code editor at the bottom shows the following Java code:

```
18         count++;
```

The system tray at the bottom right shows the date and time as 29-12-2023 15:39.

## JAVA NOTES:

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Arrays/HackthonQuestions.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Rational Application Developer interface. The main window displays a Java file named HackthonQuestions.java. The code implements a static method angryProfessor that takes two integers, k and a[], and returns "No" if the count of elements less than or equal to k is greater than or equal to k, and "YES" otherwise. It also includes a main method that reads input from System.in and prints the result to System.out. The code is annotated with Javadoc-style comments. The status bar at the bottom shows the date and time as 12/29/2023 3:42 PM.

```
1  /*HACKTHON QUESTIONS IN ARRAY*/
2
3  package Arrays;
4
5  import java.util.Scanner;
6
7  /*import java.util.Scanner; */
8
9  public class HackthonQuestions {
10
11     static String angryProfessor(int k,int a[])
12     {
13         int count=0;
14         for (int i=0;i<a.length;i++)
15         {
16             if(a[i]<=k)
17             {
18                 count++;
19
20             }
21         }
22         if(count>=k)
23         {
24             return "No";
25         }
26         else
27         {
28
29             if(count>=k)
30             {
31                 return "No";
32             }
33             else
34                 {
35                     return "YES";
36                 }
37         }
38     }
39
40     public static void main(String[] args) {
41         Scanner s=new Scanner(System.in);
42         int n=s.nextInt();
43         int k=s.nextInt();
44         int[] a=new int[n];
45         for(int i=0 ; i<a.length;i++)
46         {
47             a[i]=s.nextInt();
48         }
49         System.out.println(angryProfessor(k,a));
50     }
51 }
```

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Arrays/HackthonQuestions.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

This screenshot is identical to the one above, showing the Rational Application Developer interface with the same Java code for HackthonQuestions.java. The code implements the angryProfessor method and the main method. The status bar at the bottom shows the date and time as 12/29/2023 3:43 PM.

```
1  /*HACKTHON QUESTIONS IN ARRAY*/
2
3  package Arrays;
4
5  import java.util.Scanner;
6
7  /*import java.util.Scanner; */
8
9  public class HackthonQuestions {
10
11     static String angryProfessor(int k,int a[])
12     {
13         int count=0;
14         for (int i=0;i<a.length;i++)
15         {
16             if(a[i]<=k)
17             {
18                 count++;
19
20             }
21         }
22         if(count>=k)
23         {
24             return "No";
25         }
26         else
27         {
28             return "YES";
29         }
30     }
31
32     public static void main(String[] args) {
33         Scanner s=new Scanner(System.in);
34         int n=s.nextInt();
35         int k=s.nextInt();
36         int[] a=new int[n];
37         for(int i=0 ; i<a.length;i++)
38         {
39             a[i]=s.nextInt();
40         }
41         System.out.println(angryProfessor(k,a));
42     }
43 }
```

## JAVA NOTES:

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Arrays/HackthonQuestions.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

StringPalin... Printer.java Encapsulati... DisAdvantag... HackthonQue... PropertiesOf... >\_5

Quick Access

Console

```
11
12
13
13
14
144
5454
46
46
46
76
Exception in thread "main" java.lang.Ar
at Arrays.HackthonQuestions.mig
at Arrays.HackthonQuestions.main
```

49 //Migratory BIRD
50
51 public class HackthonQuestions {
52 static int migratoryBird (int []a) {
53 int[] birds=new int[a.length];
54 for(int i=0;i<a.length;i++) {
55 birds[a[i]]++;
56 }
57 int max=-1;
58 int maxIndex=0;
59 for(int i=0;i<birds.length;i++) {
60 if(birds[i]>max) {
61 max=birds[i];
62 maxIndex=i;
63 }
64 }
65 return maxIndex;
66 }
67 }
68
69
70
71
72
73
74
75 public static void main(String[] args) {

Writable Smart Insert 88 : 1

Type here to search

Search cognot

3:48 PM 12/29/2023

15:48 29-12-2023 ENG IN

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Arrays/HackthonQuestions.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

StringPalin... Printer.java Encapsulati... DisAdvantag... HackthonQue... PropertiesOf... >\_5

Quick Access

Console

```
11
12
13
13
14
144
5454
46
46
46
76
Exception in thread "main" java.lang.Ar
at Arrays.HackthonQuestions.mig
at Arrays.HackthonQuestions.main
```

```
73
74
75 public static void main(String[] args) {
76     Scanner s=new Scanner(System.in);
77     int n=s.nextInt();
78     int arr[]=new int[n];
79     for(int i=0;i<arr.length-1;i++){ // i<=arr.length-1 AND i<arr.length BOTH ARE SAME
80         arr[i]=s.nextInt();
81     }
82     System.out.println(migratoryBird(arr));
83
84
85
86
87
88
89 /*public class HackthonQuestions
90 {
91     static int walkWidth(int h,int [] a){
92         int width=0;
93         for(int i=0;i<a.length;i++) {
94             if(a[i]<=h)
95             {
96                 width=width+1;
97             }
98             else
99             {
```

Writable Smart Insert 88 : 1

Type here to search

Search cognot

3:49 PM 12/29/2023

15:49 29-12-2023 ENG IN

## JAVA NOTES:

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Arrays/HackthonQuestions.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Rational Application Developer interface. In the center is a code editor window displaying a Java file named HackthonQuestions.java. The code implements a method walkWidth that calculates the width of a jagged array based on its dimensions. It uses a Scanner to read input from System.in. The console window to the right shows the output of running the application, which ends with an exception message: "Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 1 at Arrays.HackthonQuestions.main at Arrays.HackthonQuestions.main". The system tray at the bottom indicates the date as 12/29/2023 and the time as 3:49 PM.

```
96         {
97             width=width+1;
98         }else
99         {
100            width=width+2;
101        }
102    }
103    return width;
104}
105public static void main(String[] args) {
106    Scanner s=new Scanner(System.in);
107    int a= s.nextInt();
108    int k= s.nextInt();
109    int []arr=new int[a];
110    for(int i=0;i<arr.length;i++)
111    {
112        arr[i]=s.nextInt();
113    }
114    System.out.println(walkWidth(k,arr));
115}
116}
117}
118}
119}
120}
121}*/
```

Console

```
<terminated> HackthonQuestions [Java Application] D:\11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 1
at Arrays.HackthonQuestions.main
at Arrays.HackthonQuestions.main
```

Type here to search

3:49 PM 12/29/2023

Search ENG IN 15:49 29-12-2023

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/JaggedArray.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

This screenshot shows the Rational Application Developer interface again. On the left is the Enterprise Explorer window, which displays a project structure for 'AMAR' containing a 'src' folder with various packages like 'src', 'Arrays', 'DAO', etc., and files such as 'D1.java', 'D1looping.java', 'D2.java', 'D3.java', 'Demo.java', 'DisAdvantagesOfArray.java', 'HackthonQuestions.java', and 'JaggedArray.java'. The code editor window in the center shows the content of 'Arrays.JaggedArray.java'. The console window on the right shows the same exception message as the previous screenshot. The system tray at the bottom shows the date as 12/29/2023 and the time as 3:54 PM.

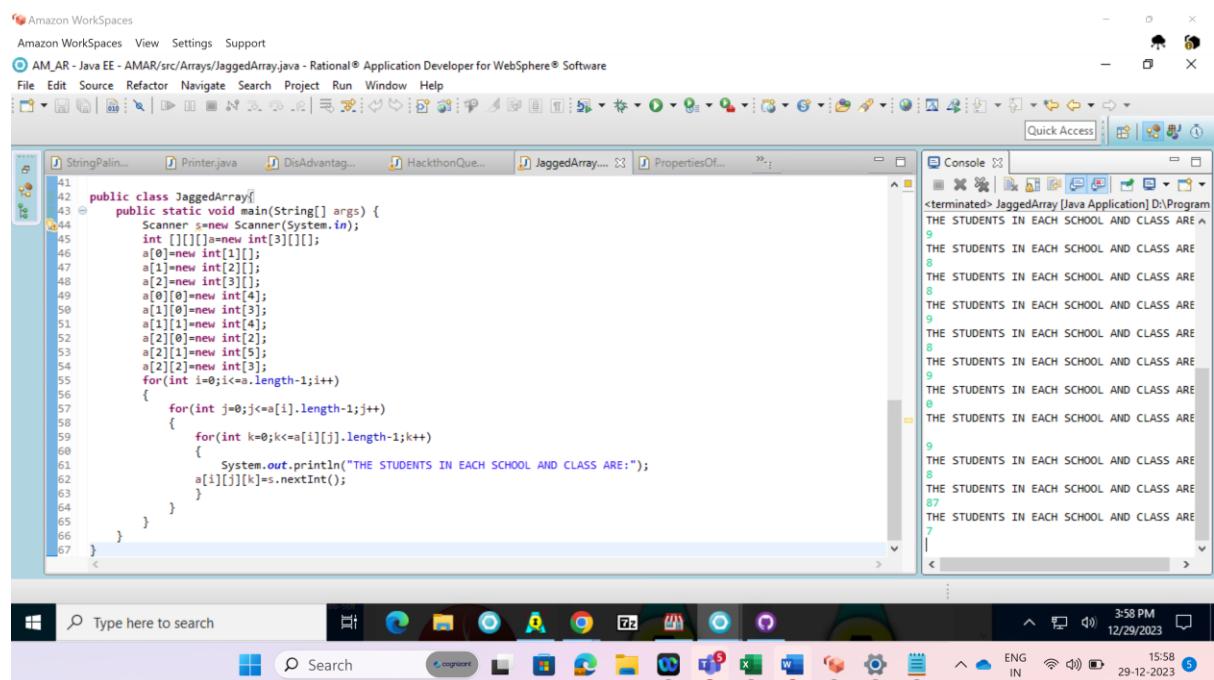
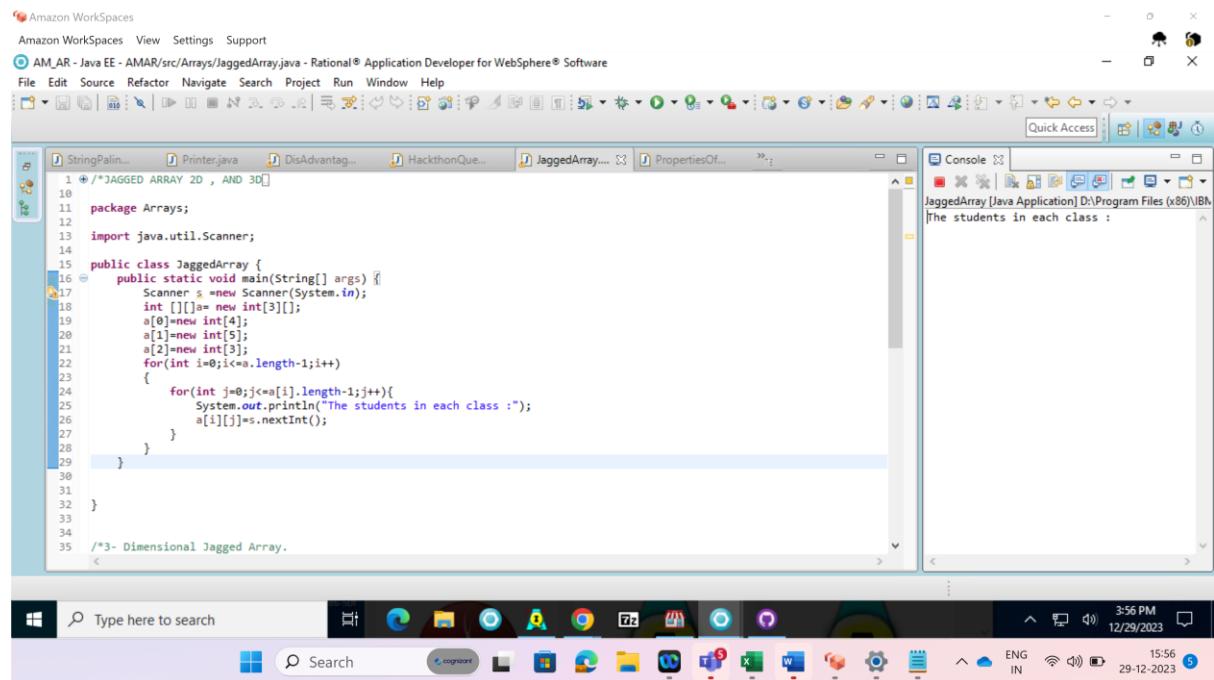
```
34
35 /*3- Dimensional Jagged Array.
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 1
at Arrays.HackthonQuestions.main
at Arrays.HackthonQuestions.main
```

Type here to search

3:54 PM 12/29/2023

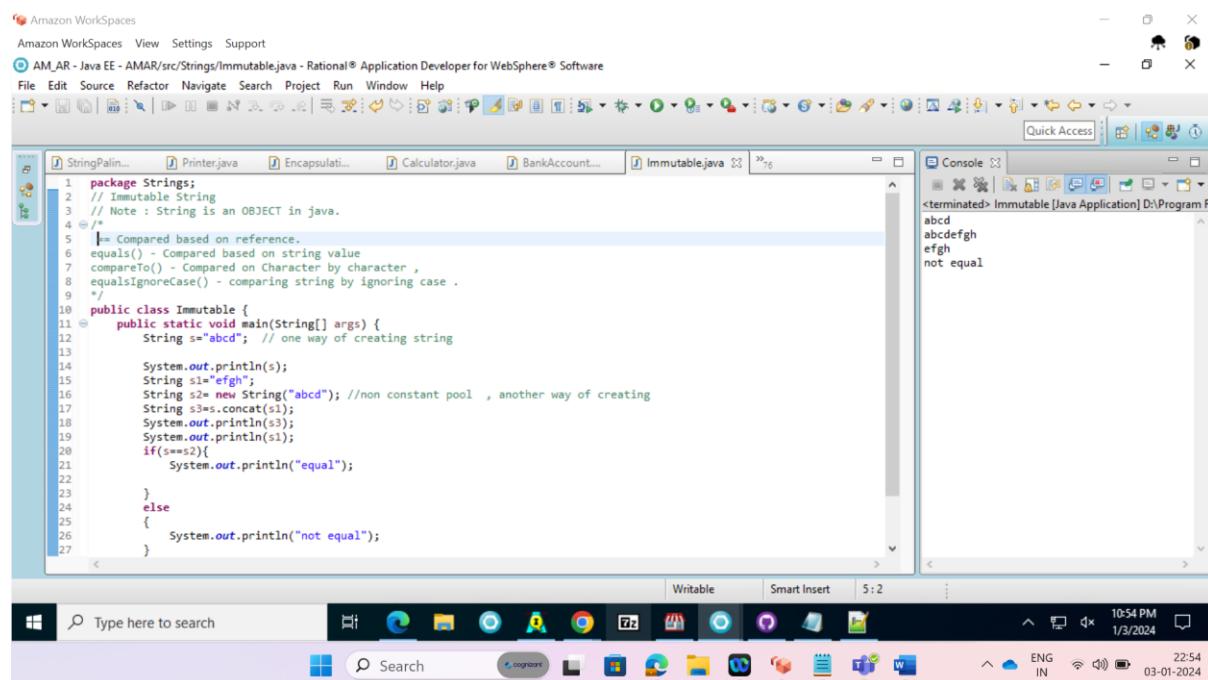
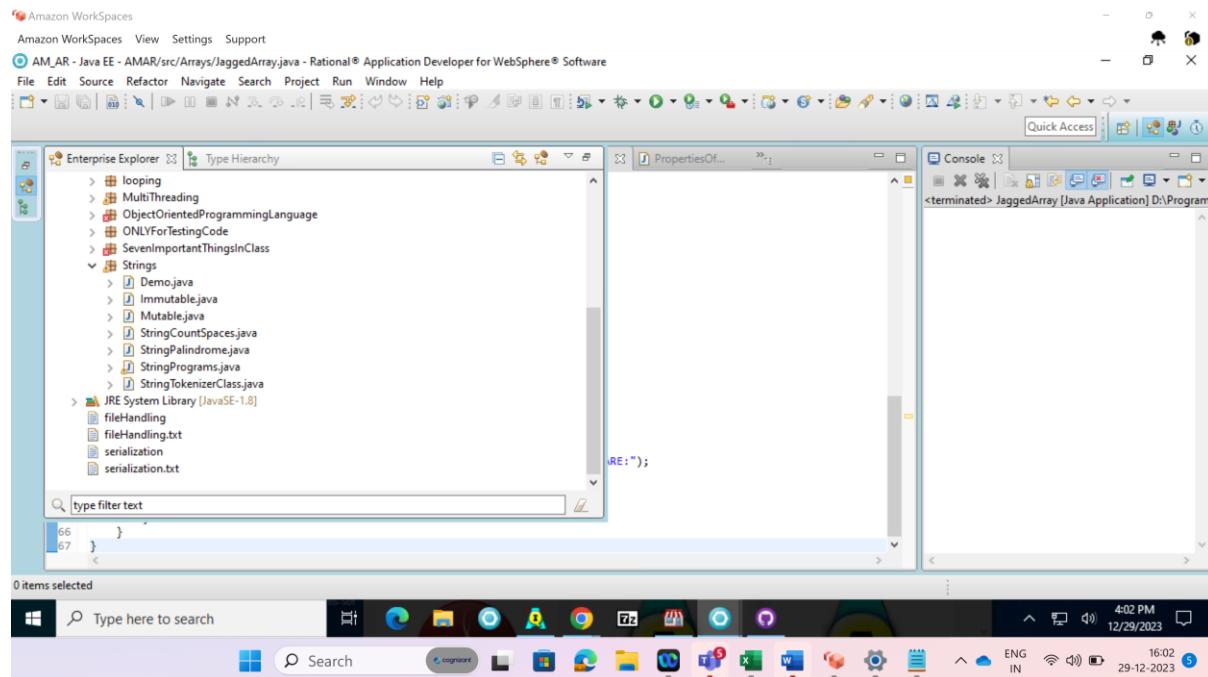
Search ENG IN 15:54 29-12-2023

JAVA NOTES:



## JAVA NOTES:

### STRINGS:



## JAVA NOTES:

The screenshot shows a Java application running in Rational Application Developer. The code in the editor compares two strings, `s1` and `s2`, using `equalsIgnoreCase` and `compareTo`. The output window shows the results of the comparison.

```
63 //public class Demo {
64     public static void main(String[] args) {
65         String s1 ="java";
66         String s2= "Java";
67         if(s1.equalsIgnoreCase(s2)){ // IT IGNORES THE CHARACTERS ONLY LOOKS THE VALUE IS SAME OR NOT
68             System.out.println("it is equal");
69         }else
70         {
71             System.out.println("not equal");
72         }
73     }
74 }
75 */
76
77 public class Demo{
78     public static void main(String[] args) {
79         String s1 ="sachin";
80         String s2="saurav";
81         System.out.println(s1.compareTo(s2));
82     }
83 }
84 
```

Console Output:

```
<terminated> Demo (2) [Java Application] D:\Program Files\AMAR\src\Strings\Demo.java
-18
```

The screenshot shows a Java application running in Rational Application Developer. The code in the editor compares two strings, `s1` and `s2`, using `equalsIgnoreCase` and `compareTo`. The output window shows the results of the comparison.

```
62 public class Demo {
63
64     public static void main(String[] args) {
65         String s1 ="java";
66         String s2= "Java";
67         if(s1.equalsIgnoreCase(s2)){ // IT IGNORES THE CHARACTERS ONLY LOOKS THE VALUE IS SAME OR NOT
68             System.out.println("it is equal");
69         }else
70         {
71             System.out.println("not equal");
72         }
73     }
74 }
75
76 /*
77 public class Demo{
78     public static void main(String[] args) {
79         String s1 ="sachin";
80         String s2="saurav";
81         System.out.println(s1.compareTo(s2));
82     }
83 */
84 
```

Console Output:

```
<terminated> Demo (2) [Java Application] D:\Program Files\AMAR\src\Strings\Demo.java
it is equal
```

## JAVA NOTES:

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Strings/Demo.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

StringPalin... Printer.java Encapsulati... Calculator.java Immutable.java Demo.java

Quick Access

Console

```
20 }
21 */
22 */
23
24
25
26 public class Demo {
27
28     public static void main(String[] args) {
29         String s1 = "java";
30         String s2 = "java";
31         if(s1.equals(s2)){
32             System.out.println("it is equal");
33         }
34         else{
35             System.out.println("not equal");
36         }
37
38     }
39
40 }
41
42
43
44
45 /*public class Demo {
46
47 }
```

Writable Smart Insert 25 : 1

Type here to search

Search cogpoint

11:02 PM 1/3/2024

ENG IN 23:02 03-01-2024

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Strings/Mutable.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

StringPalin... Printer.java Encapsulati... Calculator.java Demo.java Mutable.java

Quick Access

Console

```
1 package Strings;
2 // If some modification happens in a string is known as mutable string
3
4 // StringBuffer
5 //StringBuilder
6 public class Mutable {
7
8     public static void main(String[] args) {
9         StringBuilder st = new StringBuilder();
10        System.out.println(st.capacity());
11        st.append("java");
12
13        /*default size of the string is 16 , if incase it exceeds it'll get multiplied *2+2 with the present value */ // 16*2+2
14        st.append("javajavascriptpython");
15
16        st.trimToSize();
17        System.out.println(st);
18
19        System.out.println(st.length());
20        System.out.println(st.capacity());
21        System.out.println(st);
22        System.out.println(st.delete(0, 3));
23    }
24 }
```

Writable Smart Insert 22 : 35

Type here to search

Search cogpoint

11:09 PM 1/3/2024

ENG IN 23:09 03-01-2024

## JAVA NOTES:

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Strings/StringPalindrome.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Rational Application Developer interface. The left pane displays the Java code for `StringPalindrome` class. The right pane shows the output of the code execution in the 'Console' tab, which prints 'true'. Below the interface is a Windows taskbar with various icons and system status.

```
1 package Strings;
2
3 public class StringPalindrome {
4     static boolean palindrome(String s)
5     {
6         int i=0;
7         int j=s.length()-1;
8         while(i<j)
9         {
10             if(s.charAt(i)!=s.charAt(j))
11             {
12                 return false;
13             }
14             i++;
15             j--;
16         }
17     }
18     return true;
19 }
20
21 public static void main(String[] args) {
22     String s = "malayalam";
23     System.out.println(palindrome(s));
24 }
25
26 // TO FIND VOWELS IN A SENTENCE
```

Console

```
<terminated> StringPalindrome [Java Application]
true
```

Writable Smart Insert 25 : 6

11:15 PM 1/3/2024

Type here to search

Search

Windows Start button

System tray icons: cogpoint, battery, signal, volume, network, date/time

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Strings/StringCountSpaces.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Rational Application Developer interface. The left pane displays the Java code for `StringCountSpaces` class. The right pane shows the output of the code execution in the 'Console' tab, which prints '5'. Below the interface is a Windows taskbar with various icons and system status.

```
1 package Strings;
2
3 public class StringCountSpaces {
4     static int countSpaces(String s)
5     {
6         int count=0;
7         for(int i=0;i<s.length();i++)
8         {
9             /*THE SINGLE QUOTE IS USED TO ACCESS THE SPACES INSIDE THE STRING S */
10            if(s.charAt(i)==' ')
11            {
12                count++;
13            }
14        }
15    }
16    return count+1;
17 }
18
19 public static void main(String[] args) {
20     String s = "How The Weather Is Int?";
21     System.out.println(countSpaces(s));
22 }
23
24
25
26
27
```

Console

```
<terminated> StringCountSpaces [Java Application]
5
```

Writable Smart Insert 1:1

11:10 PM 1/3/2024

Type here to search

Search

Windows Start button

System tray icons: cogpoint, battery, signal, volume, network, date/time

## JAVA NOTES:

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Strings/StringPalindrome.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows a Java application developed in Rational Application Developer for WebSphere. The code is contained in a file named StringPalindrome.java. The code defines a static method isVowel that checks if a character is a vowel. It then uses a for loop to iterate through the characters of a string s, counting the vowels. Finally, it prints the count. The main method initializes s to "Codebuilders" and calls the isVowel method. The output window shows the result of the execution.

```
StringPalin... Printer.java Encapsulati... Immutable.java Demo.java Mutable.java StringCount... >76
```

```
50 // Accessing the character without printing vowels.
51
52 static String isVowel(String s , String t )
53 {
54
55     for(int i=0;i<s.length();i++)
56     {
57         char k = s.charAt(i);
58         if(k=='a' || k=='e' || k=='i' || k=='o' || k=='A' || k=='E' || k=='I' || k=='O' || k=='U')
59         {
60
61         }
62         else{
63             t=t+k;
64         }
65     }
66     return t;
67 }
68
69 public static void main(String[] args) {
70     String s="Codebuilders";
71     String t="";
72     System.out.println(isVowel(s,t));
73 }
74 }
```

Console > terminated> StringPalindrome [Java Application] Cdbuldrns

Writable Smart Insert 50 : 1

Type here to search 11:13 PM 1/3/2024

Search ENG IN 23:13 03-01-2024

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Strings/StringPalindrome.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows a Java application developed in Rational Application Developer for WebSphere. The code is contained in a file named StringPalindrome.java. The code defines a static method isVowel that counts the vowels in a given string. It then uses a for loop to iterate through the characters of a string s, counting the vowels. Finally, it prints the count. The main method initializes s to "Codebuilders" and calls the isVowel method. The output window shows the result of the execution.

```
StringPalin... Printer.java Encapsulati... Immutable.java Demo.java Mutable.java StringCount... >76
```

```
24 */
25
26 // TO FIND VOWELS IN A SENTENCE
27
28 static int isVowel(String s)
29 {
30     int count=0;
31     for(int i=0;i<s.length();i++)
32     {
33         char k = s.charAt(i);
34         if(k=='a' || k=='e' || k=='i' || k=='o' || k=='A' || k=='E' || k=='I' || k=='O' || k=='U')
35         {
36             count++;
37         }
38     }
39     return count;
40 }
41
42 public static void main(String[] args) {
43     String s="Codebuilders";
44     System.out.println(isVowel(s));
45 }
```

Console > terminated> StringPalindrome [Java Application] 4

Writable Smart Insert 75 : 9

Type here to search 11:15 PM 1/3/2024

Search ENG IN 23:15 03-01-2024

## JAVA NOTES:

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Strings/StringPrograms.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Rational Application Developer interface. The main window displays a Java file named StringPrograms.java. The code implements a static method reverseString that takes two strings as parameters and returns a reversed copy of the first string. It uses a for loop to iterate from the end of the string to the beginning, concatenating each character into a new string. The main method demonstrates this by printing the reverse of the string "HOW IS". The console output shows the result: "SI MOH". The taskbar at the bottom shows various application icons and the date/time as 11:18 PM 1/3/2024.

```
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
```

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Strings/StringPrograms.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Rational Application Developer interface. The main window displays a Java file named StringPrograms.java. The code defines a class with a static method countchar that counts the occurrences of a specified character in a given string. It iterates through the string and increments a counter whenever it finds the target character. The main method tests this with the string "How is the weather" and the character 'h'. The console output shows the result: "1". The taskbar at the bottom shows various application icons and the date/time as 11:20 PM 1/3/2024.

```
32 /*TO COUNT THE NUMBER OF CHARACTERS AND PRINT IT EG =>H  O/P= 2*/
33
34 public class StringPrograms
35 {
36     static int countchar(String s,String a)
37     {
38         int count=0;
39         char key ='H';
40
41         for(int i=0;i<s.length()-1;i++)
42         {
43             if(key==s.charAt(i))
44             {
45                 count++;
46             }
47
48         }
49         return count;
50     }
51
52     public static void main(String[] args) {
53         String s = " How is the weather ";
54         String a="h";
55         System.out.println(countchar(s,a));
56     }
57
58 }
```

## JAVA NOTES:

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Strings/StringPrograms.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Rational Application Developer interface. The left pane displays a Java code editor with the following code:

```
58
59 // Palindrome
60 static String reverseString(String s, String a)
61 {
62     for(int i=s.length()-1;i>=0;i--)
63     {
64         a+=s.charAt(i);
65     }
66     return a;
67 }
68 static boolean palindrome(String s, String a)
69 {
70     if(s.equals(reverseString(s,a)))
71     {
72         return true;
73     }
74     return false;
75 }
76 public static void main(String[] args) {
77     String s ="HOW IS ";
78     String a="";
79     System.out.println(reverseString(s,a));
80 }
81 }
```

The right pane shows a console output window with the text "SI WOH". Below the application window is a Windows taskbar with various icons and system status.

Amazon WorkSpaces

Amazon WorkSpaces View Settings Support

AM\_AR - Java EE - AMAR/src/Strings/StringTokenizerClass.java - Rational® Application Developer for WebSphere® Software

File Edit Source Refactor Navigate Search Project Run Window Help

The screenshot shows the Rational Application Developer interface. The left pane displays a Java code editor with the following code:

```
1 package Strings;
2
3 import java.util.StringTokenizer;
4
5 public class StringTokenizerClass {
6
7     public static void main(String[] args) {
8
9
10        /*String s ="JAVA PYTHON STRING";
11        StringTokenizer st=new StringTokenizer(s);
12        while(st.hasMoreTokens())
13        {
14            System.out.println(st.nextToken());
15        }*/
16        String s ="He as t eball";
17        StringTokenizer st=new StringTokenizer(s , "h");
18        while(st.hasMoreTokens())
19        {
20            System.out.println(st.nextToken());
21        }
22    }
23
24 }
25
26 }
```

The right pane shows a console output window with the text "He as t eball". Below the application window is a Windows taskbar with various icons and system status.

## JAVA NOTES:

The screenshot shows a Java application running in Rational Application Developer for WebSphere. The code in the editor is as follows:

```
1 package Strings;
2
3 import java.util.StringTokenizer;
4
5 public class StringTokenizerClass {
6     public static void main(String[] args) {
7
8         String s ="JAVA PYTHON STRING";
9         StringTokenizer st=new StringTokenizer(s);
10        while(st.hasMoreTokens())
11        {
12            System.out.println(st.nextToken());
13        }
14
15        /* String s ="Hehas theball";
16        StringTokenizer st=new StringTokenizer(s , "h");
17        while(st.hasMoreTokens())
18        {
19            System.out.println(st.nextToken());
20        }
21        */
22    }
23 }
24
25 }
26 }
27 }
```

The console output shows the tokens: JAVA, PYTHON, STRING.

Below the IDE is a Windows taskbar with various pinned icons and system status.

## OOPS :

The screenshot shows a Java application running in Rational Application Developer for WebSphere. The code in the editor is as follows:

```
1 package ObjectOrientedProgrammingLanguage;
2
3 > Abstraction1.java
4 > Abstraction.java
5 > AccessModifier.java
6 > Aggregation1composition1.java
7 > AggregationComposition.java
8 > BankAccount.java
9 > Calculator.java
10 > ConstructorOverloading.java
11 > Customer.java
12 > Demo1.java
13 > Encapsulation.java
14 > FinalKeyword.java
15 > Inheritance.java
16 > InheritancesTYPES.java
17 > Interface.java
18 > Interface1.java
19 > MethodOverloading.java
20 > Polymorphism.java
21
22
23
24
25
26
27 }
```

The Enterprise Explorer view shows a tree structure of Java files under the package ObjectOrientedProgrammingLanguage. The console output shows the tokens: JAVA, PYTHON, STRING.

Below the IDE is a Windows taskbar with various pinned icons and system status.

## JAVA NOTES:

### METHOD OVERLOADING:

The screenshot shows the Rational Application Developer interface. The main window displays a Java file named 'MethodOverloading.java' with the following code:

```
1 package ObjectOrientedProgrammingLanguage;
2
3 /*SOLVING REAL TIME PROBLEMS IS OOPS
4 *
5 *Object = Property (datatypes) , Behaviour(methods)
6 */
7
8 //We can create multiple methods with same name in method overloading
9
10 // Calculator.java
11
12 public class MethodOverloading {
13
14 }
```

The status bar at the bottom indicates the date and time as 11:28 PM on 1/3/2024.

The screenshot shows the Rational Application Developer interface. The main window displays a Java file named 'Printer.java' with the following code:

```
6 public class Printer {
7
8     void print(int a)
9     {
10         System.out.println(a);
11     }
12     void print(int a, int b){
13         System.out.println(a + " " + b);
14     }
15     void print(float a)
16     {
17         System.out.println(a);
18     }
19     void print(double a)
20     {
21         System.out.println(a);
22     }
23     void print(String a)
24     {
25         System.out.println(a);
26     }
27
28     public static void main(String[] args) {
29         Printer p=new Printer();
30         p.print(9, 8);
31         p.print(7.3f);
32     }
33 }
```

The status bar at the bottom indicates the date and time as 11:34 PM on 1/3/2024.

JAVA NOTES:

The screenshot shows a Rational Application Developer for WebSphere Software interface. The left pane displays the source code for `Calculator.java` in a code editor. The code defines a `Calculator` class with methods for addition and multiplication of integers and doubles. The right pane shows the `Console` output for a terminated Java application, displaying two printed values: 37.8 and 12.7.

```
1 package ObjectOrientedProgrammingLanguage;
2 // METHOD OVERLOADING
3
4 // type of parameter, no of parameter , order of parameter
5
6 public class Calculator {
7     int add(int a , int b)
8     {
9         return a+b;
10    }
11    double add(int a , double b)
12    {
13        return a+b;
14    }
15    double main(double a , int b)
16    {
17
18        return a*b;
19    }
20    public static void main(String[] args) {
21        Calculator c=new Calculator();
22        System.out.println(c.add(18, 19.8));
23        System.out.println(c.main(12.7, 1));
24    }
25
26 }
```

JAVA NOTES:

## ACCESS SPECIFIERS:

## EXCEPTION HANDLING:

	A	B
1	<b>RUN TIME ERROR</b>	<b>EXCEPTION</b>
2	It occurs during runtime	Occurs during runtime
3	abrupt termination happens	abrupt termination happens
4		
5	Runtime error happens becoz of <b>faulty coding</b>	Exception happens becoz of <b>faulty input</b>
6		
7	Run time error <b>CANNOT</b> be handled using try-catch	Exception error <b>CAN</b> be handled using try-catch
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

JAVA NOTES: