

LAB NO 1

DATA STRUCTURES AND ALGORITHMS

OBJECTIVE: To study the concepts of String Constant Pool, String literals, String immutability and Wrapper classes.

TASK NO 1:

Write a program that initialize five different strings using all the above mentioned ways, i.e., a) string literals b) new keyword also use intern method and show string immutability.

INPUT:

```
package shaheer.javaaid;






public class ShaheerJavaid {

    public static void main(String[] args) {
        String str1 = "Hello Shaheer";
        String str2 = "Hello Javaid";
        String str3 = new String("Hello Moosa");
        String str4 = str1.intern();
        System.out.println(str1);
        System.out.println(str2);
        System.out.println(str3);
        System.out.println(str4);
        String original = "Immutable";
        String modified = original.concat("String");
        System.out.println("Original: " + original);
        System.out.println("Modified: " + modified);
    }

}
```

OUTPUT:

Output - Shaheer Javaid (run) X

 run:
 Hello Shaheer
 Hello Javaid
 Hello Moosa
 Hello Shaheer
Original: Immutable
Modified: ImmutableString
BUILD SUCCESSFUL (total time: 0 seconds)**TASK NO 2:**





Write a program to convert primitive data type Double into its respective wrapper object.

INPUT:

```
package shaheer.javaaid;  
public class ShaheerJavaid {  
  
    public static void main(String[] args) {  
        double primitiveDouble = 60.56;  
        Double wrapperDouble = primitiveDouble;  
        System.out.println("Primitive Double : " + primitiveDouble);  
        System.out.println("Wrapper Double: " + wrapperDouble);  
    }  
}
```

OUTPUT:

Output - Shaheer Javaid (run) X

 run:
 Primitive Double : 60.56
 Wrapper Double: 60.56
 BUILD SUCCESSFUL (total time: 0 seconds)

TASK NO 3:

Write a program that initialize five different strings and perform the following operations. a. Concatenate all five strings. b. Convert fourth string to uppercase. c. Find the substring from the concatenated string from 8 to onward.

INPUT:

```
package shaheer.javaaid;
public class ShaheerJavaid {

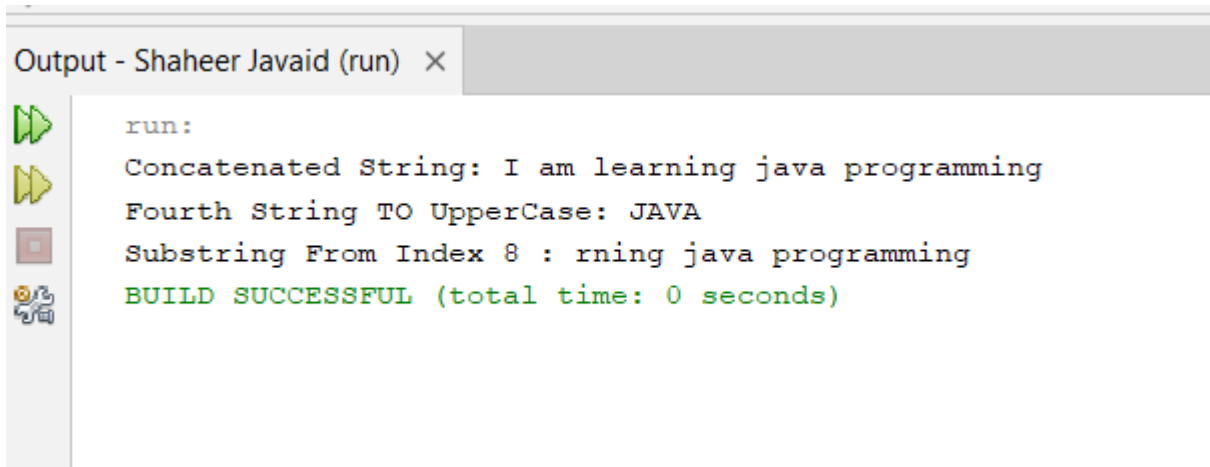
    public static void main(String[] args) {
        String str1 = "I ";
        String str2 = "am ";
        String str3 = "learning ";
        String str4 = "java ";
        String str5 = "programming ";

        // (A)
        String concatenatedString = str1 + str2 + str3 + str4 + str5;
        System.out.println("Concatenated String: " + concatenatedString);

        // (B)
        String str4Upper = str4.toUpperCase();
        System.out.println("Fourth String TO UpperCase: " + str4Upper);

        // (C)
        String subString = concatenatedString.substring(8);
        System.out.println("Substring From Index 8 : " + subString);
    }
}
```

OUTPUT:



```
Output - Shaheer Javaid (run) X
run:
Concatenated String: I am learning java programming
Fourth String TO UpperCase: JAVA
Substring From Index 8 : rning java programming
BUILD SUCCESSFUL (total time: 0 seconds)
```

TASK NO 4:

You are given two strings word1 and word2. Merge the strings by adding letters in alternating order, starting with word1. If a string is longer than the other, append the additional letters onto the end of the merged string. Return the merged string. Example: Input: word1 = "abc", word2 = "pqr" Output: "apbqcr" Explanation: The merged string will be merged as so: word1: a b c word2: p q r merged: a p b q c r

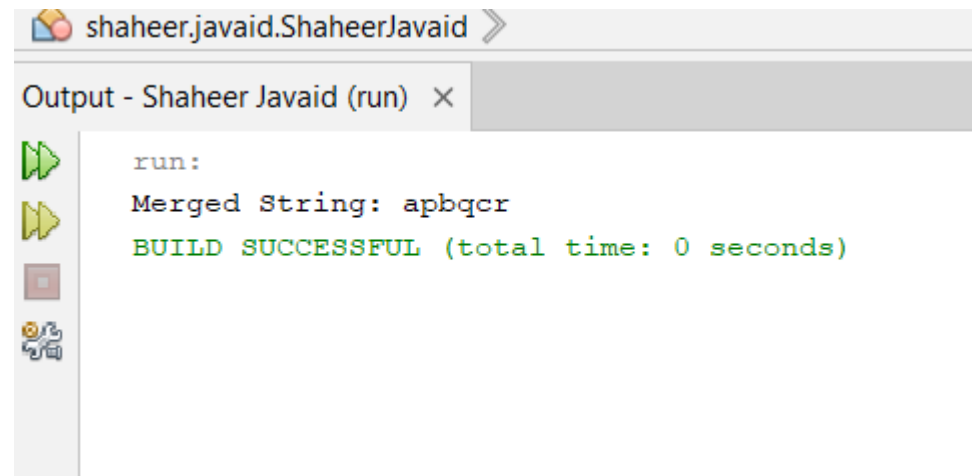
INPUT:

```
package shaheer.javaaid;
public class ShaheerJavaid {

    public static String mergeAlternately(String word1, String word2) {
        StringBuilder merged = new StringBuilder();
        int maxLength = Math.max(word1.length(), word2.length());

        for (int i = 0; i < maxLength; i++) {
            if (i < word1.length()) {
                merged.append(word1.charAt(i));
            }
            if (i < word2.length()) {
                merged.append(word2.charAt(i));
            }
        }
        return merged.toString();
    }

    public static void main(String[] args) {
        String word1 = "abc";
        String word2 = "pqr";
        System.out.println("Merged String: " + mergeAlternately(word1, word2));
    }
}
```

OUTPUT:

The screenshot shows an IDE interface. At the top, a tab is labeled 'shaheer.javaide.ShaheerJavaid' with a right-pointing arrow. Below it, a window titled 'Output - Shaheer Javaid (run)' is open. On the left side of this window is a vertical toolbar with icons for running (a green play button), stepping through code (a yellow play button), stopping (a red square), and debugging (a blue bug icon). The main area of the output window displays the following text:

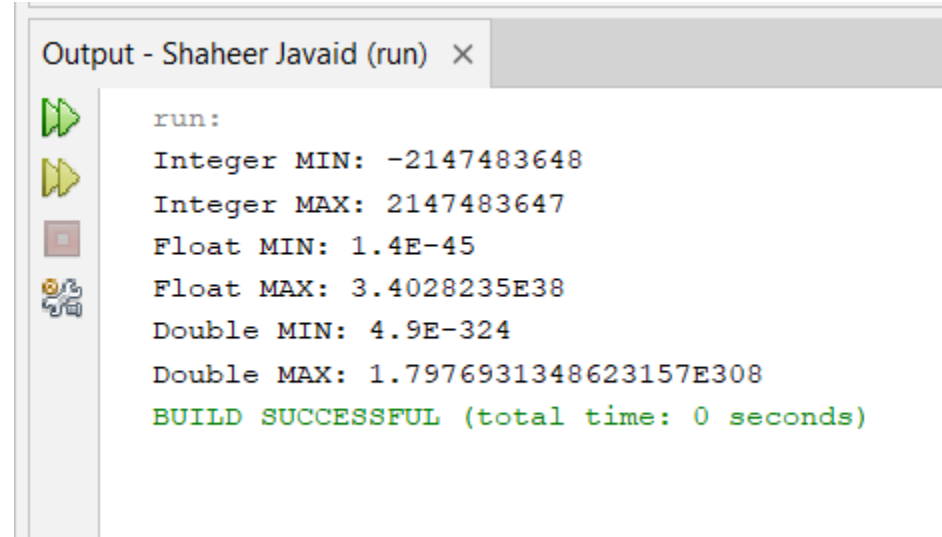
```
run:
Merged String: apbqcr
BUILD SUCCESSFUL (total time: 0 seconds)
```

TASK NO 5

Write a Java program to find the minimum and maximum values of Integer, Float, and Double using the respective wrapper class constants.

INPUT:

```
package shaheer.javaaid;  
  
public class ShaheerJavaid {  
  
    public static void main(String[] args) {  
        System.out.println("Integer MIN: " + Integer.MIN_VALUE);  
        System.out.println("Integer MAX: " + Integer.MAX_VALUE);  
        System.out.println("Float MIN: " + Float.MIN_VALUE);  
        System.out.println("Float MAX: " + Float.MAX_VALUE);  
        System.out.println("Double MIN: " + Double.MIN_VALUE);  
        System.out.println("Double MAX: " + Double.MAX_VALUE);  
    }  
}
```

OUTPUT:

```
run:  
Integer MIN: -2147483648  
Integer MAX: 2147483647  
Float MIN: 1.4E-45  
Float MAX: 3.4028235E38  
Double MIN: 4.9E-324  
Double MAX: 1.7976931348623157E308  
BUILD SUCCESSFUL (total time: 0 seconds)
```

:

HOME TASKS

TASK NO 1:

Write a JAVA program to perform Autoboxing and also implement different methods of wrapper class.

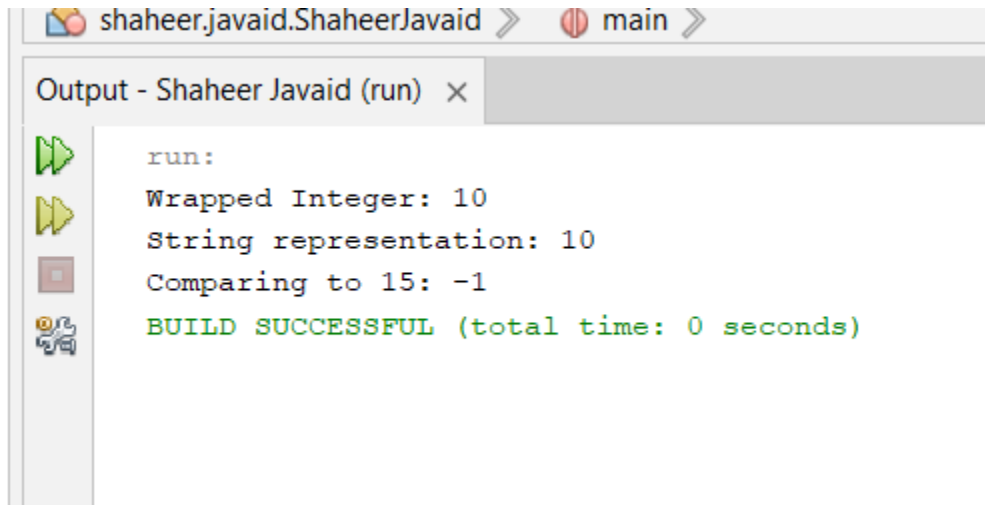
INPUT:

```
package shaheer.javaaid;
public class ShaheerJavaid {

    public static void main(String[] args) {
        int num = 10;
        Integer wrappedNum = num; // Autoboxing
        System.out.println("Wrapped Integer: " + wrappedNum);

        // Using wrapper class methods
        System.out.println("String representation: " + wrappedNum.toString());
        System.out.println("Comparing to 15: " + wrappedNum.compareTo(15));
    }
}
```

OUTPUT:



```
run:
Wrapped Integer: 10
String representation: 10
Comparing to 15: -1
BUILD SUCCESSFUL (total time: 0 seconds)
```

TASK NO 2

Write a Java program to count the number of even and odd digits in a given integer using Autoboxing and Unboxing.

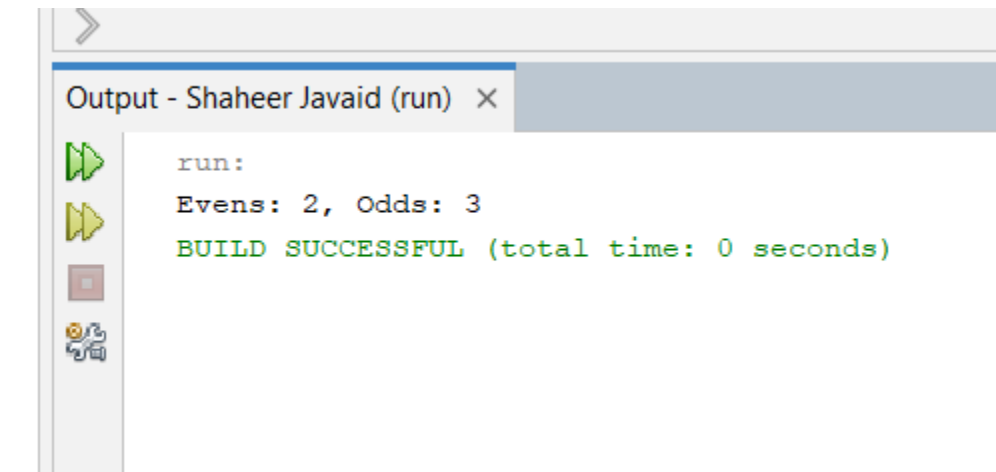
INPUT:

```
package shaheer.javaaid;
public class ShaheerJavaid {

    public static void main(String[] args) {
        Integer num = 12345;
        int evenCount = 0, oddCount = 0;

        while (num != 0) {
            int digit = num % 10;
            if (digit % 2 == 0) {
                evenCount++;
            } else {
                oddCount++;
            }
            num /= 10;
        }

        System.out.println("Evens: " + evenCount + ", Odds: " + oddCount);
    }
}
```

OUTPUT:

```
run:
Evens: 2, Odds: 3
BUILD SUCCESSFUL (total time: 0 seconds)
```

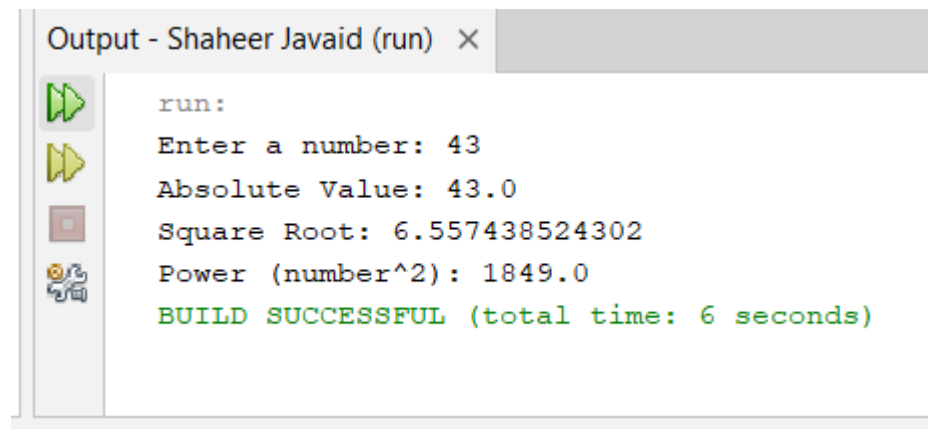
TASK NO 3:

Write a Java program to find the absolute value, square root, and power of a number using Math class methods, while utilizing Autoboxing and Wrapper classes.

INPUT:

:

```
package shaheer.javaaid;  
import java.util.Scanner;  
public class ShaheerJavaid {  
  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print("Enter a number: ");  
        double num = scanner.nextDouble();  
  
        System.out.println("Absolute Value: " + Math.abs(num));  
        System.out.println("Square Root: " + Math.sqrt(num));  
        System.out.println("Power (number^2): " + Math.pow(num, 2));  
    }  
}
```

OUTPUT:

```
run:  
Enter a number: 43  
Absolute Value: 43.0  
Square Root: 6.557438524302  
Power (number^2): 1849.0  
BUILD SUCCESSFUL (total time: 6 seconds)
```

TASK NO 4

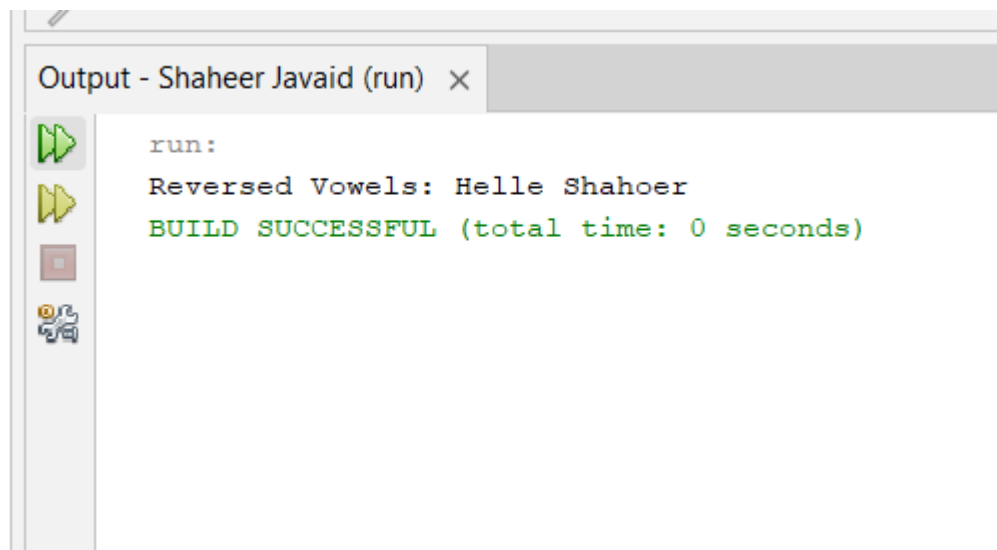
Write a Java program to reverse only the vowels in a string.

INPUT:

```
package shaheer.javaid;

public class ShaheerJavaid {
    public static void main(String[] args) {
        String str = "Hello Shaheer";
        char[] chars = str.toCharArray();
        int left = 0, right = str.length() - 1;
        while (left < right) {
            if (isVowel(chars[left]) && isVowel(chars[right])) {
                char temp = chars[left];
                chars[left] = chars[right];
                chars[right] = temp;
                left++;
                right--;
            } else {
                if (!isVowel(chars[left])) left++;
                if (!isVowel(chars[right])) right--;
            }
        }
        System.out.println("Reversed Vowels: " + new String(chars));
    }

    private static boolean isVowel(char ch) {
        return "AEIOUaeiou".indexOf(ch) != -1;
    }
}
```

OUTPUT:

```
Output - Shaheer Javaid (run) x
run:
Reversed Vowels: Helle Shahoer
BUILD SUCCESSFUL (total time: 0 seconds)
```

TASK NO 5:

Write a Java program to find the longest word in a sentence

:

INPUT:

```
package shaheer.javaaid;

public class ShaheerJavaid {
    public static void main(String[] args) {
        String sentence = "Java is bit difficult to learn";
        String[] words = sentence.split(" ");
        String longestWord = "";

        for (String word : words) {
            if (word.length() > longestWord.length()) {
                longestWord = word;
            }
        }

        System.out.println("Longest Word: " + longestWord);
    }
}
```

OUTPUT:

```
ut - Shaheer Javaid (run) x
run:
Longest Word: difficult
BUILD SUCCESSFUL (total time: 0 seconds)
|
```