NAME: ANEEQ SHAMS ROLL NO: BSE-230 SECTION: E

Lab #01

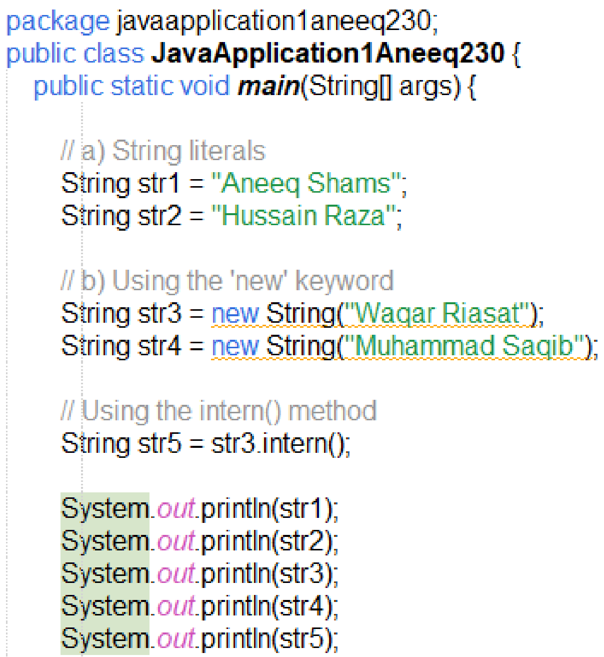
“Data Structures and Algorithms”

Exercises

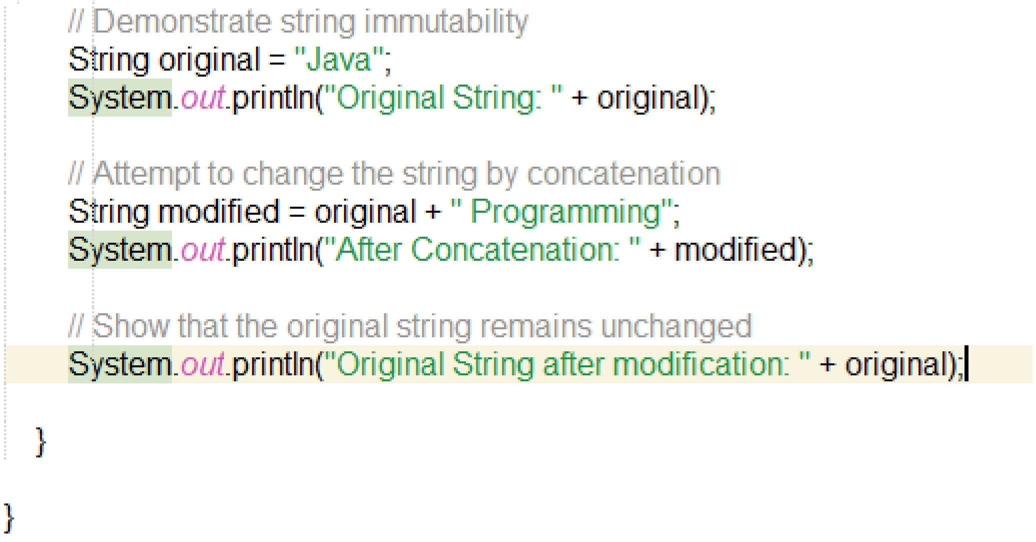
Lab Tasks:

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

- Input:

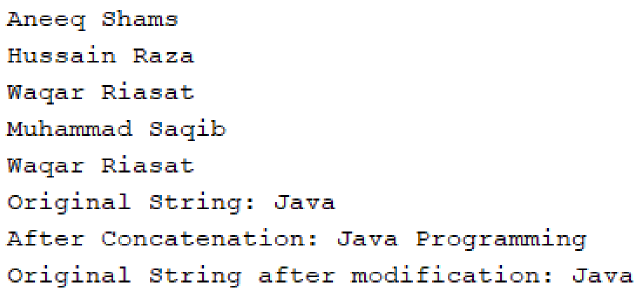


```
package javaapplication1aneeq230;
public class JavaApplication1Aneeq230 {
    public static void main(String[] args) {

        // a) String literals
        String str1 = "Aneeq Shams";
        String str2 = "Hussain Raza";

        // b) Using the 'new' keyword
        String str3 = new String("Waqar Riasat");
        String str4 = new String("Muhammad Saqib");

        // Using the intern() method
        String str5 = str3.intern();

        System.out.println(str1);
        System.out.println(str2);
        System.out.println(str3);
        System.out.println(str4);
        System.out.println(str5);
```



```
        // Demonstrate string immutability
        String original = "Java";
        System.out.println("Original String: " + original);

        // Attempt to change the string by concatenation
        String modified = original + " Programming";
        System.out.println("After Concatenation: " + modified);

        // Show that the original string remains unchanged
        System.out.println("Original String after modification: " + original);

    }

}
```

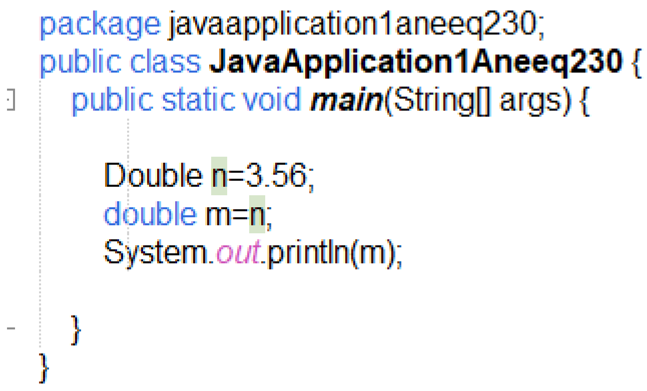
Output:



```
Aneeq Shams
Hussain Raza
Waqar Riasat
Muhammad Saqib
Waqar Riasat
Original String: Java
After Concatenation: Java Programming
Original String after modification: Java
```

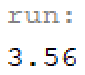
2. Write a program to convert primitive data type Double into its respective wrapper object.

- Input:



```
package javaapplication1aneeq230;
public class JavaApplication1Aneeq230 {
    public static void main(String[] args) {

        Double n=3.56;
        double m=n;
        System.out.println(m);

    }
}
```

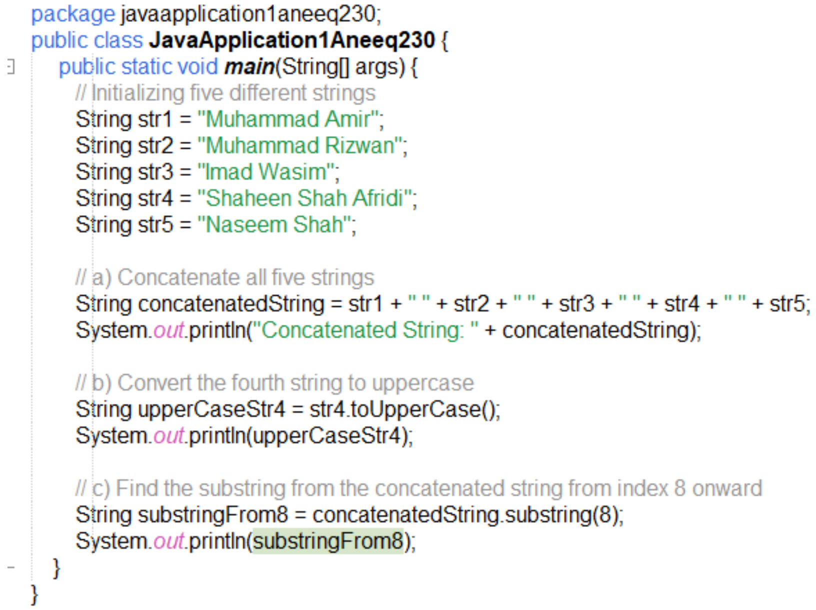
Output:



```
run:
3.56
```

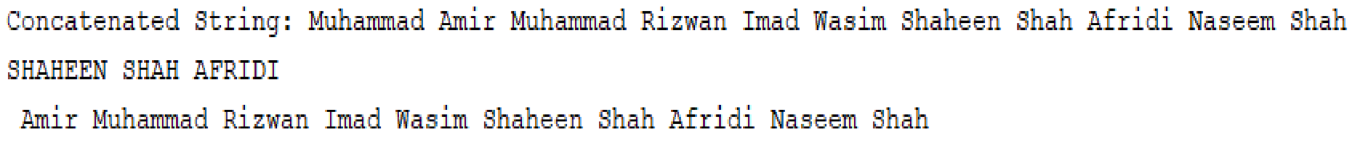
3. Write a program that initialize five different strings and perform the following operations.
   a. Concatenate all five stings.
   b. Convert fourth string to uppercase.
   c. Find the substring from the concatenated string from 8 to onward

- Input:



```
package javaapplication1aneeq230;
public class JavaApplication1Aneeq230 {
    public static void main(String[] args) {
      // Initializing five different strings
      String str1 = "Muhammad Amir";
      String str2 = "Muhammad Rizwan";
      String str3 = "Imad Wasim";
      String str4 = "Shaheen Shah Afridi";
      String str5 = "Naseem Shah";

      // a) Concatenate all five strings
      String concatenatedString = str1 + " " + str2 + " " + str3 + " " + str4 + " " + str5;
      System.out.println("Concatenated String: " + concatenatedString);

      // b) Convert the fourth string to uppercase
      String upperCaseStr4 = str4.toUpperCase();
      System.out.println(upperCaseStr4);

      // c) Find the substring from the concatenated string from index 8 onward
      String substringFrom8 = concatenatedString.substring(8);
      System.out.println(substringFrom8);
    }
}
```

Output:



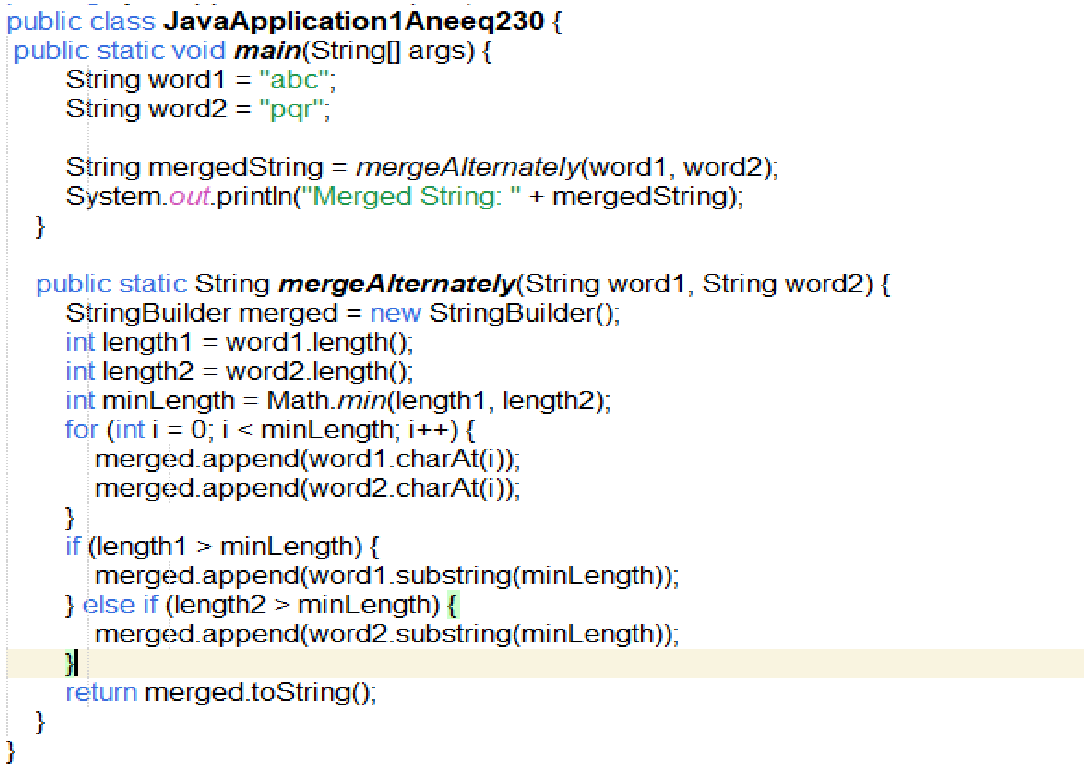
```
Concatenated String: Muhammad Amir Muhammad Rizwan Imad Wasim Shaheen Shah Afridi Naseem Shah
SHAHEEN SHAH AFRIDI
 Amir Muhammad Rizwan Imad Wasim Shaheen Shah Afridi Naseem Shah
```

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"

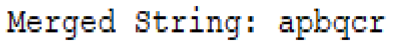
   Explanation: The merged string will be merged as so:
   word1: a b c
   word2: p q r
   merged: a p b q c r

- Input:



```
public class JavaApplication1Aneeq230 {
 public static void main(String[] args) {
      String word1 = "abc";
      String word2 = "pqr";

      String mergedString = mergeAlternately(word1, word2);
      System.out.println("Merged String: " + mergedString);
   }

   public static String mergeAlternately(String word1, String word2) {
      StringBuilder merged = new StringBuilder();
      int length1 = word1.length();
      int length2 = word2.length();
      int minLength = Math.min(length1, length2);
      for (int i = 0; i < minLength; i++) {
         merged.append(word1.charAt(i));
         merged.append(word2.charAt(i));
      }
      if (length1 > minLength) {
         merged.append(word1.substring(minLength));
      } else if (length2 > minLength) {
         merged.append(word2.substring(minLength));
      }
      return merged.toString();
   }
}
```

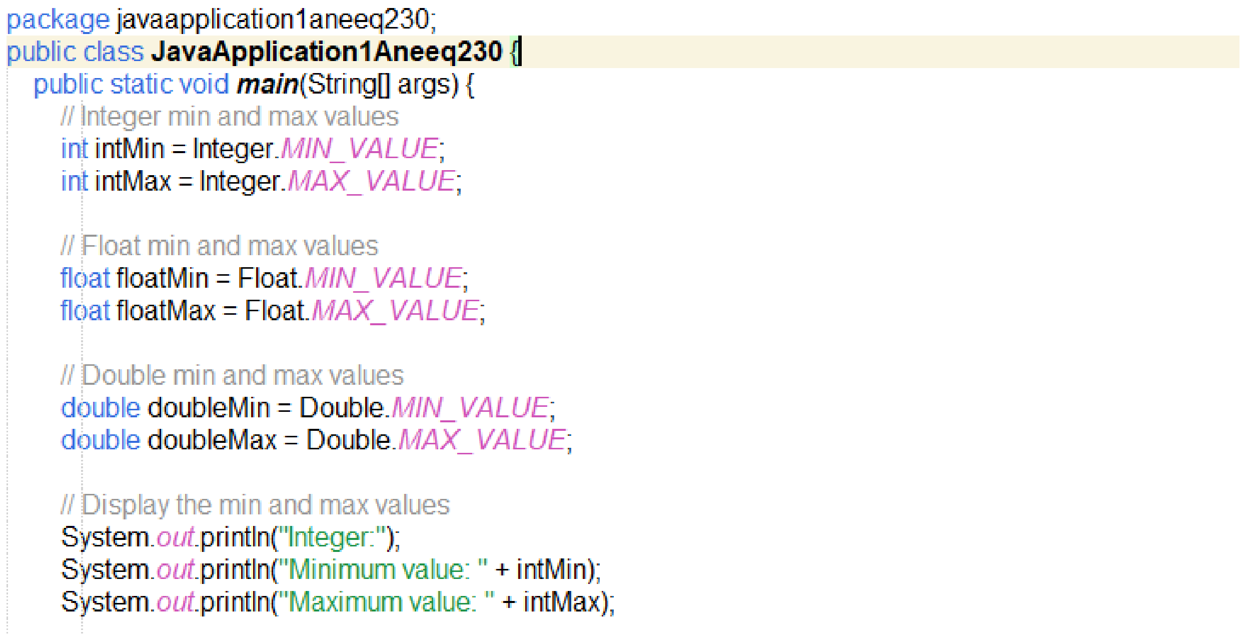
Output:



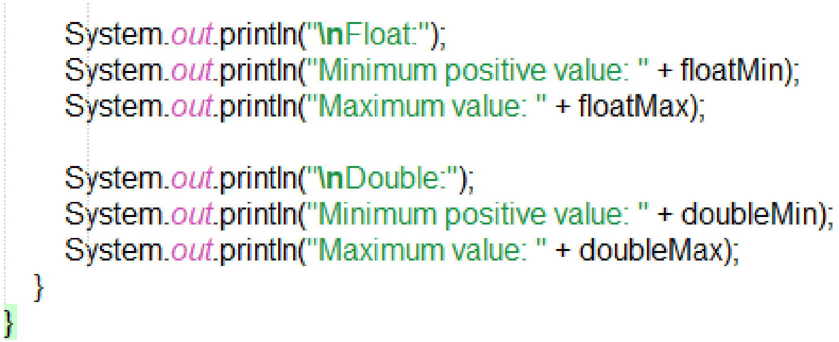
```
Merged String: apbqcr
```

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

- Input:

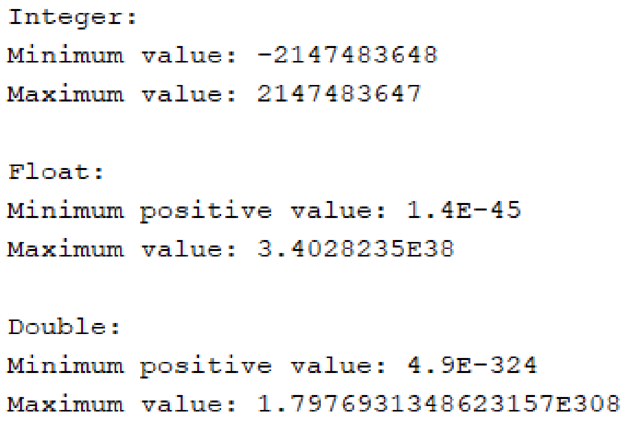


```
package javaapplication1aneeq230;
public class JavaApplication1Aneeq230 {
   public static void main(String[] args) {
      // Integer min and max values
      int intMin = Integer.MIN_VALUE;
      int intMax = Integer.MAX_VALUE;

      // Float min and max values
      float floatMin = Float.MIN_VALUE;
      float floatMax = Float.MAX_VALUE;

      // Double min and max values
      double doubleMin = Double.MIN_VALUE;
      double doubleMax = Double.MAX_VALUE;

      // Display the min and max values
      System.out.println("Integer:");
      System.out.println("Minimum value: " + intMin);
      System.out.println("Maximum value: " + intMax);
```



```
        System.out.println("\nFloat:");
        System.out.println("Minimum positive value: " + floatMin);
        System.out.println("Maximum value: " + floatMax);

        System.out.println("\nDouble:");
        System.out.println("Minimum positive value: " + doubleMin);
        System.out.println("Maximum value: " + doubleMax);
    }
}
```

Output:

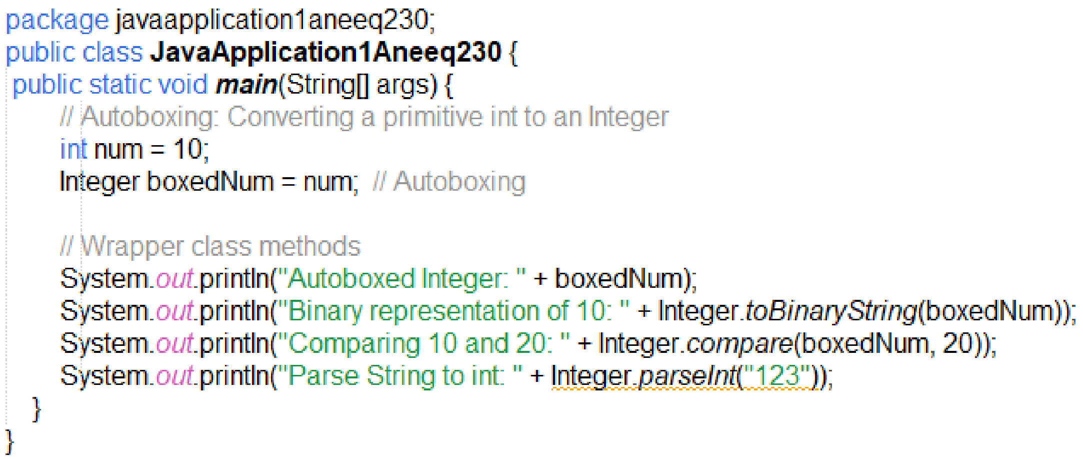


```
Integer:
Minimum value: -2147483648
Maximum value: 2147483647

Float:
Minimum positive value: 1.4E-45
Maximum value: 3.4028235E38

Double:
Minimum positive value: 4.9E-324
Maximum value: 1.7976931348623157E308
```

Home Tasks:

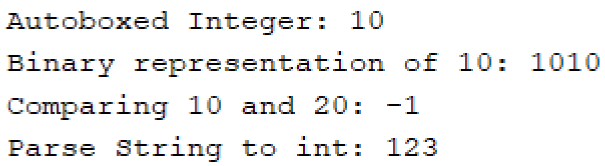
1. Write a JAVA program to perform Autoboxing and also implement different methods of wrapper class.

- Input:



```
package javaapplication1aneeq230;
public class JavaApplication1Aneeq230 {
 public static void main(String[] args) {
        // Autoboxing: Converting a primitive int to an Integer
        int num = 10;
        Integer boxedNum = num;  // Autoboxing

        // Wrapper class methods
        System.out.println("Autoboxed Integer: " + boxedNum);
        System.out.println("Binary representation of 10: " + Integer.toBinaryString(boxedNum));
        System.out.println("Comparing 10 and 20: " + Integer.compare(boxedNum, 20));
        System.out.println("Parse String to int: " + Integer.parseInt("123"));
    }
}
```

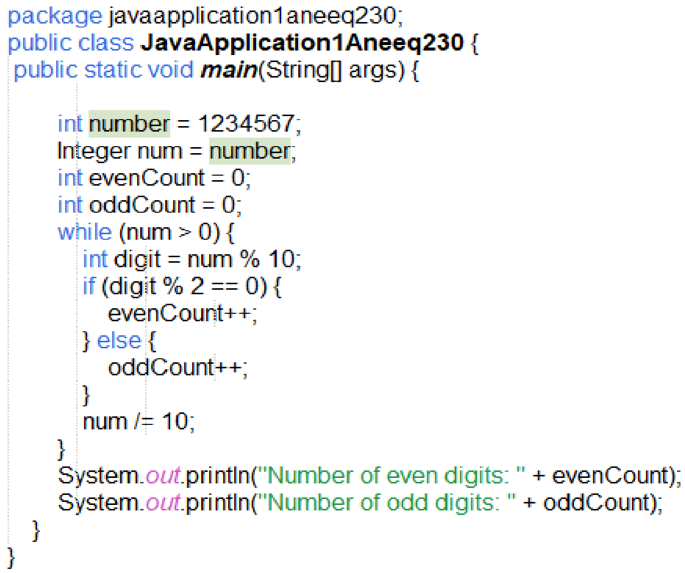
Output:



```
Autoboxed Integer: 10
Binary representation of 10: 1010
Comparing 10 and 20: -1
Parse String to int: 123
```

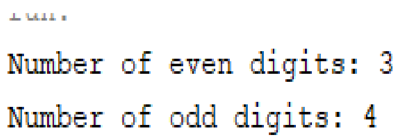
2. Write a Java program to count the number of even and odd digits in a given integer using Autoboxing and Unboxing.

- Input:



```
package javaapplication1aneeq230;
public class JavaApplication1Aneeq230 {
 public static void main(String[] args) {

     int number = 1234567;
     Integer num = number;
     int evenCount = 0;
     int oddCount = 0;
     while (num > 0) {
        int digit = num % 10;
        if (digit % 2 == 0) {
            evenCount++;
        } else {
            oddCount++;
        }
        num /= 10;
     }
     System.out.println("Number of even digits: " + evenCount);
     System.out.println("Number of odd digits: " + oddCount);
  }
}
```

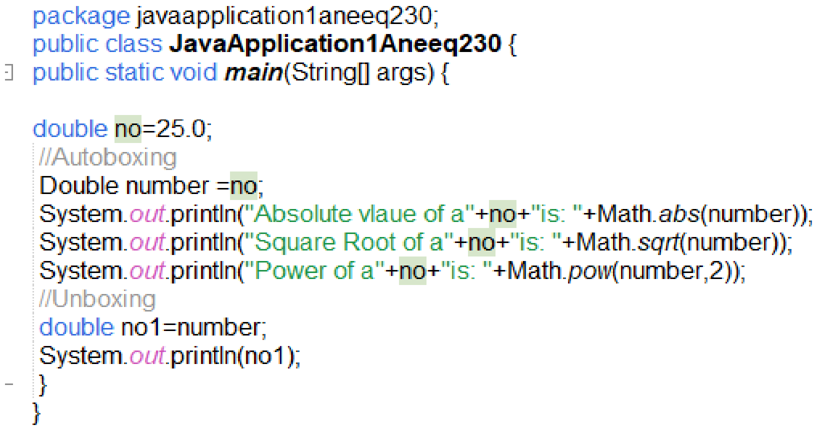
Output:



```
run:
Number of even digits: 3
Number of odd digits: 4
```

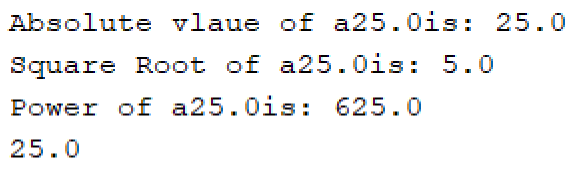
3. Write a Java program to find the absolute value, square root, and power of a number using Math class methods, while utilizing Autoboxing and Wrapper classes.

- Input:



```
package javaapplication1aneeq230;
public class JavaApplication1Aneeq230 {
public static void main(String[] args) {

double no=25.0;
 //Autoboxing
 Double number =no;
 System.out.println("Absolute vlaue of a"+no+"is: "+Math.abs(number));
 System.out.println("Square Root of a"+no+"is: "+Math.sqrt(number));
 System.out.println("Power of a"+no+"is: "+Math.pow(number,2));
 //Unboxing
 double no1=number;
 System.out.println(no1);
 }
}
```

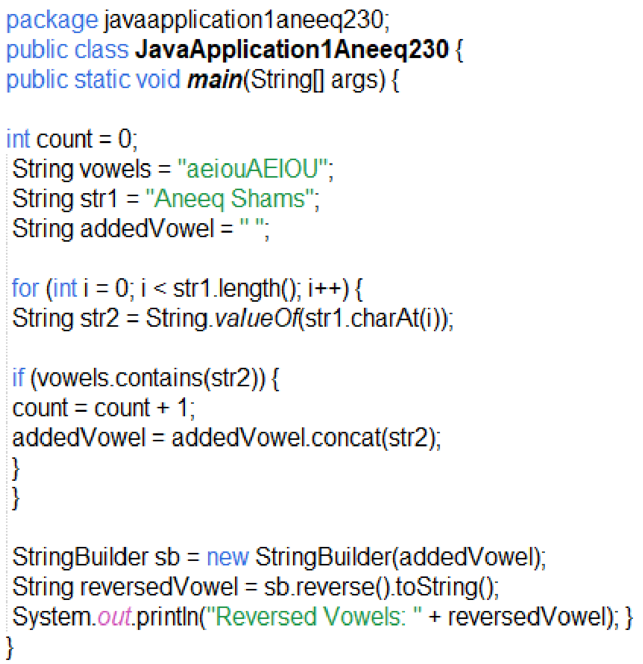
Output:



```
Absolute vlaue of a25.0is: 25.0
Square Root of a25.0is: 5.0
Power of a25.0is: 625.0
25.0
```

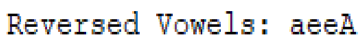
4. Write a Java program to reverse only the vowels in a string.

- Input:



```
package javaapplication1aneeq230;
public class JavaApplication1Aneeq230 {
public static void main(String[] args) {

int count = 0;
 String vowels = "aeiouAEIOU";
 String str1 = "Aneeq Shams";
 String addedVowel = " ";

 for (int i = 0; i < str1.length(); i++) {
 String str2 = String.valueOf(str1.charAt(i));

 if (vowels.contains(str2)) {
 count = count + 1;
 addedVowel = addedVowel.concat(str2);
 }
 }

 StringBuilder sb = new StringBuilder(addedVowel);
 String reversedVowel = sb.reverse().toString();
 System.out.println("Reversed Vowels: " + reversedVowel); }
}
```

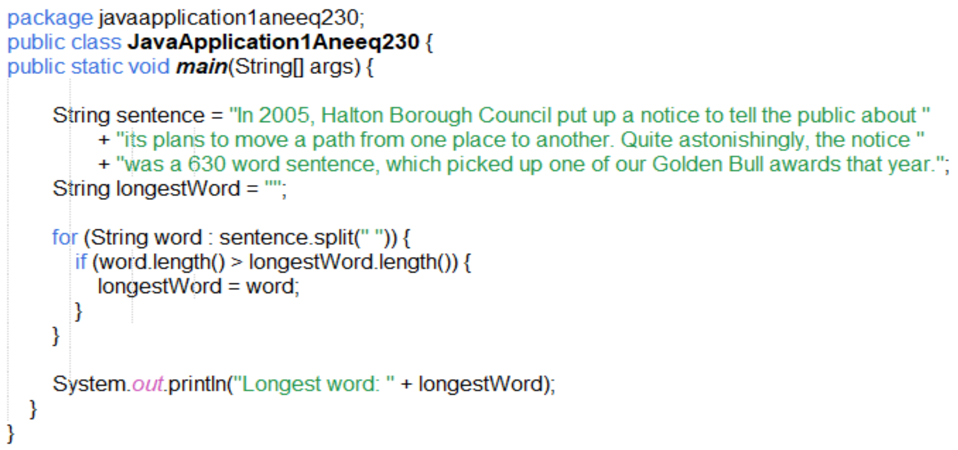
Output:



```
Reversed Vowels: aeeA
```

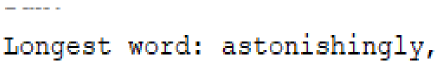
5. Write a Java program to find the longest word in a sentence.

- Input:



```
package javaapplication1aneeq230;
public class JavaApplication1Aneeq230 {
public static void main(String[] args) {

    String sentence = "In 2005, Halton Borough Council put up a notice to tell the public about "
        + "its plans to move a path from one place to another. Quite astonishingly, the notice "
        + "was a 630 word sentence, which picked up one of our Golden Bull awards that year.";
    String longestWord = "";

    for (String word : sentence.split(" ")) {
      if (word.length() > longestWord.length()) {
        longestWord = word;
      }
    }

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

Output:



```
Longest word: astonishingly,
```