INHERITANCE

```
interface BankInterest{
 void getRateOfInterest();
 }

 class BankA implements BankInterest{
 public void getRateOfInterest(){
    System.out.println("Rate of interest of Bank A is 6%");
   }
  }

 class BankB implements BankInterest{
  public void getRateOfInterest(){
   System.out.println("Rate of Interest of Bank B is 5%");
  }
 }
 class BankC implements BankInterest{
  public void getRateOfInterest(){
   System.out.println("Rate of Interest of Bank C is 8%");
  }
 }

 class Bank{
 public static void main(String args[]){
  BankA a=new BankA();
  a.getRateOfInterest();
  BankB b= new BankB();
  b.getRateOfInterest();
  BankC c=new BankC();
  c.getRateOfInterest();
  }
  }
```

STRING TOKENIZER

```
import java.util.StringTokenizer;

public class TokenizerExample {
   public static void main(String[] args) {

      String str = "Java is a programming language";

      StringTokenizer tokenizer = new StringTokenizer(str, " ");

      System.out.println("Number of tokens: " + tokenizer.countTokens());

      while (tokenizer.hasMoreTokens()) {
         System.out.println(tokenizer.nextToken());
      }
   }
}
```

BINARY SEARCH

```
import java.util.Arrays;
import java.util.Scanner;

class BinarySearchExample {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      int item, flag = 0, m, i, low, mid;
      int[] a = new int[100];

      System.out.println("ENTER THE max size:");
      m = sc.nextInt();

      System.out.println("ENTER THE ELEMENTS OF ARRAY:");
      for (i = 0; i < m; i++) {
         a[i] = sc.nextInt();
      }

      Arrays.sort(a, 0, m);

      System.out.println("ENTER THE ELEMENT TO BE SEARCHED:");
      item = sc.nextInt();

      low = 0;
      int high = m - 1;

      while (low <= high) {
         mid = (low + high) / 2;

         if (a[mid] == item) {
            System.out.println("The element is found at index " + mid);
            flag = 1;
            break;
         } else if (item < a[mid]) {
            high = mid - 1;
         } else {
            low = mid + 1;
         }
      }

      if (flag == 0) {
         System.out.println("Element not found");
      }

      sc.close();
   }
}
```

THREAD SYNCHRONIZATION

```
import java.util.*;

class Display {
  synchronized void printMessage(String msg) {
    System.out.print("[" + msg);
    try {
      Thread.sleep(1000);
    } catch (InterruptedException e) {
      System.out.print(e);
    }
    System.out.print("]");
  }
}

class MessageTask implements Runnable {
  private Display display;
  private String message;

  public MessageTask(Display display, String message) {
    this.display = display;
    this.message = message;
  }

  public void run() {
    display.printMessage(message);
  }
}

public class Main {
  public static void main(String[] args) {
    Display display = new Display();
    Thread t1 = new Thread(new MessageTask(display, "Hello"));
    Thread t2 = new Thread(new MessageTask(display, "World"));
    t1.start();
    t2.start();
  }
}
```

```
import java.util.Scanner;

public class PalindromeExample {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      System.out.print("Enter a string: ");
      String str = sc.nextLine();

      if (isPalindrome(str)) {
         System.out.println("The string is a palindrome.");
      } else {
         System.out.println("The string is not a palindrome.");
      }

      sc.close();
   }

   public static boolean isPalindrome(String str) {
      int left = 0;
      int right = str.length() - 1;

      while (left < right) {
         if (str.charAt(left) != str.charAt(right)) {
            return false;
         }
         left++;
         right--;
      }
      return true;
   }
}
```

```
import java.util.Scanner;

public class CharacterFrequency {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      System.out.print("Enter a string: ");
      String str = sc.nextLine();

      System.out.print("Enter a character to find its frequency: ");
      char ch = sc.next().charAt(0);

      int frequency = getFrequency(str, ch);
      System.out.println("Frequency of '" + ch + "' in the string: " + frequency);

      sc.close();
   }

   public static int getFrequency(String str, char ch) {
      int count = 0;

      for (int i = 0; i < str.length(); i++) {
         if (str.charAt(i) == ch) {
            count++;
         }
      }
      return count;
   }
}
```