**Assignments**

1. **Identify 5 real life objects in the context of software for online marketplace like flipkart, amazon**
2. Shopping cart
3. Review
4. order
5. List
6. User
7. **Identify 5 attributes and 2 methods / behaviours for each of the below objects - Student, Laptop, Door, Printer , Chair, Web Browser, Lift.**
8. Student
   1. Attribute

* Name
* Age
* Height
* Weight
* Address

1. method

* study()
* Scoremarks()

1. Laptop
   1. Attribute

* Brand
* Model
* Color
* Processor
* Ram

1. method

* doprogram()
* playgame()

1. Door
   1. Attribute

* Color
* Type
* Height
* Width

1. Method

* open()
* close()

1. Printer
   1. Attribute

* Brand
* Model
* Type
* Color
* weight

1. method

* print()

1. Chair
   1. Attribute

* Type
* Color
* Height
* material

1. method

* sit()
* isfoldable()

1. Web Browser
   1. Attribute

* Version
* Bookmark
* History
* Company
* speed

1. method

* deleteHistory()
* addBookmark()

1. Lift
   1. Attribute

* Company
* Capacity
* Maintenance status

1. method

* open()
* close()

1. **Identify which all attributes can be initialized in constructor for each of below objects - Student, Laptop, Door, Printer , Chair, Web Browser, Lift**
2. Student

* Name
* Age
* marks

1. Laptop

* Brand
* Model
* processor

1. Door

* Color
* Size
* material

1. Printer

* Brand
* Model

1. Chair

* Color
* type

1. Web browser

* Version
* Bookmark
* history

**Assignments**

Write and execute below programs in Eclipse

1. **Print “Hello Automation Testing”**

**public** **class** **print** **{**

**public** **static** **void** **main(String[]** **args){**

**System.out.print(“Hello Automation Testing”);**

**}**

**}**

1. **Create Student Object initialize all attributes in constructor and print all values within print() method inside object.**

**public** **class** **student** **{**

**public** **String** **name;**

**public** **int** **age;**

**public** **student(String** **name,int** **age){**

***this*.name=name;**

***this*.age=age;**

**}**

**public** **void** **print(){**

**System.out.println("name "+name);**

**System.out.println("age "+age);**

**}**

**public** **static** **void** **main(String[]** **args){**

**student s=*new*** **student("saksham",** **22);**

**s.print();**

**}**

**}**

1. **Create Laptop Object. Initialize few attributes in constructor (without passing arguments). Assign remaining values through setter methods of object. Print all attributes.**

**public** **class** **laptop** **{**

**public** **String** **brand;**

**public** **String** **model;**

**public** **laptop(){}**

**public** **void** **setBrand(String** **brand){**

***this*.brand=brand;**

**}**

**public** **void** **setModel(String** **model){**

***this*.model=model;**

**}**

**public** **void** **print(){**

**System.out.println("brand "+brand);**

**System.out.println("model "+model);**

**}**

**public** **static** **void** **main(String[]** **args){**

**laptop l =*new*** **laptop();**

**l.setBrand("DEll");**

**l.setModel("Vostro");**

**l.print();**

**}**

**}**

1. **Create Printer object. Initialize few attributes in constructor (by passing arguments). Assign remaining values through setter methods of object. Print all attributes.**

**public** **class** **printer** **{**

**public** **String** **brand;**

**public** **String** **color;**

**public** **printer(String** **brand,** **String** **color){**

***this*.brand=brand;**

***this*.color=color;**

**}**

**public** **void** **print(){**

**System.out.println("brand "+brand);**

**System.out.println("color "+color);**

**}**

**public** **static** **void** **main(String[]** **args){**

**printer l =*new*** **printer("hp","black");**

**l.print();**

**}**

**}**

1. **Create any object and have another object as one of the attribute inside first object. Assign values in constructor as well as setter methods for attributes of both objects and print all values.**

**public** **class** **employee** **{**

**String** **name;**

**String** **age;**

**public** **employee(String** **name,** **String** **age){**

***this*.name=name;**

***this*.age=age;**

**}**

**public** **employee(employee** **obj){**

**name=obj.name;**

**age=obj.age;**

**}**

**public** **void** **print(){**

**System.out.println("name "+name);**

**System.out.println("age "+age);**

**}**

**public** **static** **void** **main(String[]** **args){**

**employee l =*new*** **employee("saksham","22");**

**employee l1=*new*** **employee(l);**

**l.print();**

**l1.print();**

**}**

**}**

**Encapsulation**

1. What is Encapsulation

* Bundling / wrapping od data and method within a class
* Allowing access to the data by using methods and providing data security

2. Why Encapsulation is required/useful in Java?

* To protect data security
* To enable abstraction
* To enhance code flexibility and maintainiability

3. Are there any errors in below program? Find all (if any) and list according to line numbers

class EncapsulationDemo{

private String empName;

public int empAge;

public String getEmpName(){

return empName;

}

public int getEmpAge(){

return empAge;

}

 public void setEmpAge(int newValue){

empAge = newValue;

}

public void setEmpName(String newValue){

empName = newValue;

}

}

public class EncapsTest{

public static void main(String args[]){

EncapsulationDemo obj = new EncapsulationDemo();

obj.setEmpName("Ajay");

obj.setEmpAge(32);

 System.out.println("Employee Name: " + obj.getEmpName());    // error

System.out.println("Employee Age: " + obj.getEmpAge());

obj.empName = "Smita";

obj.empAge = 33;

System.out.println("New Employee Name: " + obj.empName);  //error

 System.out.println("New Employee Age: " + obj.empAge);

}

 }

4. Write a program to demonstrate use of encapsulation and explain what issues we would

have faced in absence of encapsulation.

**Inheritance**

1. What is Inheritance?

* It is mechanism that allow a class to inherit properties and behaviours from another class
* It enable code reuse
* Create hierarchical relationships b/w classes

2. Why Inheritance is required/useful in Java?

* To promote code reusability
* Maintain code structure
* Help to use polymoripism

3. Write a java program with Objects Car, RemoteControlledCar, Toy and explain how

inheritance helps here.

public class car {

    String model;

    String brand;

    public car(String model,String brand){

*this*.model=model;

*this*.brand=brand;

    }

}

 class RemoteControlledCar extends car{

    String color;

    public RemoteControlledCar(String color, String model,String brand){

*super*(model,brand);

*this*.color=color;

    }

 }

    class toy extends RemoteControlledCar{

    String type;

    public toy(String type, String color,String model,String brand){

*super*(model,brand,color);

*this*.type=type;

    }

    public void print(){

        System.out.println("type "+type);

        System.out.println("color "+color);

        System.out.println("model "+model);

        System.out.println("brand "+brand);

    }

    public static void main(String args[]) {

    String type="plastic";

    String color="grey";

    String model="phantom";

    String brand="RR";

    toy t=*new* toy(type,color,model,brand);

    t.print();

    }

}

4. Write a program to inherit below Person in your Employee class. Employee object will

have empid and deptname. Create object of Employee class and print all values - name,

empid and depltname. Assume suitable data for Employee object. DO NOT modify

Person class.

public class Person {

String name = ;

public Person (String name) {

this.name = name;

}

}

public class Employee {

    String empid;

    String deptname;

    public Employee(String empid,String deptname){

*this*.empid=empid;

*this*.deptname=deptname;

    }

}

 class Person extends Employee{

    String name;

    public Person(String name, String empid,String deptname){

*super*(empid,deptname);

*this*.name=name;

    }

    public static void main(String args[]) {

    String name="Saksham";

    String emp="GS-3933";

    String dept="Corp";

    Person P=*new* Person(name,emp,dept);

    System.out.println(P);

}

}

**Polymorphism**

1. What is polymorphism (Runtime and Compile time)?

Ability of an object to take multiple forms

Run-time:-

* It is having the samemethid with same parameter but associated with different class
* The call is not resolved by complier

Compile time:-

* In this one or more method share the same name with didderenr parameter and different run time.
* The call is resolved by complier

2. Why polymorphism is required/useful in Java?

* Promote code resusability
* To write code that can work with the object of different types without knowing their specific implementation
* To customise the bahaviour of inherited method to suit the specific need of that each subclass

3. Is Inheritance must for Polymorphism in java? Explain.

No , inheritance is not neccessary as polymorphism can be acheived through both inheritance and interface.

4. Demonstrate polymorphism with java program where run() method for Dog, Cat, Mouse

shows use of polymorphism. Assume suitable data.

public class Animal {

    public void run(){

        System.out.println("The animal makes a sound");

    }

}

class Dog extends Animal {

  public void run() {

    System.out.println("Dog run");

  }

}

class Cat extends Animal {

  public void run() {

    System.out.println("Cat run");

  }

}

class Mouse extends Animal {

  public void run() {

    System.out.println("Mouse run");

  }

}

class Polymorphism{

    public static void main(String[] args) {

        Animal a=*new* Animal();

        Animal d=*new* Dog();

        Animal c=*new* Cat();

        Animal m=*new* Mouse();

        a.run();

        d.run();

        c.run();

        m.run();

    }

}

**Abstraction**

1. What is Abstraction?

It is way to hide the internal details and complexity of an object and provide a simplified.

2. Why Abstraction is required/useful in Java?

* Simplify the complexity bt hiding the complex implementation of some detail of program
* Resuability of class behavious
* It enables you to write generic and reusability code that can adapt differenr functionality pf class

3. Create abstract class Printer and use this in another class DeskjetPrinter and

LaserPrinter and use method print() to demonstrate Abstraction

abstract public class printer {

    abstract void print();

}

class DeskjetPrinter extends printer{

    void print(){

        System.out.println("print");

    }

}

class LaserPrinter extends printer{

    void print(){

        System.out.println("print");

    }

    public static void main(String[] args){

    printer p1=*new* DeskjetPrinter();

    printer p2=*new* LaserPrinter();

    p1.print();

    p2.print();

    }

}

4. Is it useful to extend multiple classes? Explain.

* It lead to diamond problem and make problem more complex as compiler not able to decide which method it should choose or inherit   
    
  **6. Assignment**

Write a program to read attached txt file (sampleFile.txt) below and give counts of

1. # of words
2. # of unique words
3. # of palindrome words
4. # of vowels
5. # of articles (a,an,the)
6. Reverse entire file content word by word
7. While reversing, remove all the articles(a,an, the)
8. While reversing, Introduce “#10” before every 10 words
9. While reversing, Insert “#UNIQUE before every unique word you have identified

public class string\_ex{

    static void CountWord(String str){

        System.out.println("count word = " + str.length());

    }

    static void countPalindrome(String str){

        int count=0;

        String words[]=str.split("\\s");

*for* (String i*:*words){

*if*(isPalindrome(i)){

                count++;

            }

        }

        System.out.println(count);

    }

    static boolean isPalindrome(String i){

        int left=0;

        int right=i.length()-1;

*while*(left<right){

*if*(i.charAt(left)!=i.charAt(right)){

*return* false;

            }

*else*{

                left++;right--;

            }

        }

*return* true;

    }

    static void CountAll(String str){

        int v = 0,c = 0;

*for*(int i=0;i<str.length();i++){

*if*(str.charAt(i)=='a' || str.charAt(i)=='e' || str.charAt(i)=='i' || str.charAt(i)=='o' ||str.charAt(i)=='u' ){v++;}

*else*{c++;}

        }

        String[] s=str.split("//s");

        int a=0,an=0,the=0;

*for* (String word*:*s){

*if*(word.equalsIgnoreCase("a")){

                a++;

            }

*else* *if*(word.equalsIgnoreCase("an")){

                an++;

            }

*else* *if*(word.equalsIgnoreCase("the")){

                the++;

            }

        }

        System.out.println("Vowels = " + v + " Consonents = " + c);

        System.out.println("number of a's = " + a + " number of an's = " + an + " number of the's = " + the +" ");

    }

    static void ReverseWord(String str){

        String words[]=str.split("\\s");

        String reverseWord="";

*for*(String w*:*words){

            StringBuilder sb=*new* StringBuilder(w);

            sb.reverse();

            reverseWord+=sb.toString()+" ";

        }

        System.out.println("Reverse Word = "+reverseWord.trim());

    }

    static String reverse(String str){

        String ReverseStr="";

*for*(int i=str.length()-1;i>=0;i--){

            ReverseStr+=str.charAt(i);

        }

*return* ReverseStr;

    }

    static void ReverseAndReplaceArticle(String str){

        String words[]=str.split("\\s");

        String new\_str="";

*for*(String i*:*words){

*if*(!(i.equals("a") || i.equals("an")|| i.equals("the"))){

                new\_str+=i+" ";

            }

        }

        System.out.println("Remove Article while Reversing:- " + reverse(new\_str));

    }

    static void ReverseAndIntroduce10(String str){

         String words[]=str.split("\\s");

         String d = "";

*for*(int i=0;i<words.length;i++){

*if*(i%10==0){

                d =d+ "01# ";

            }

           d=d+words[i] +" ";

         }

         System.out.println("Add #10:- "+reverse(d));

    }

    static void ReverseAndIntroduce1Unique(String str){

         String words[]=str.split("\\s");

         String newStr="";

*for*(int i=0;i<words.length;i++){

            int count=1;

*for*(int j=i+1;j<words.length;j++){

*if*(words[i].equalsIgnoreCase(words[j])){

                    count++;

                }

            }

*if*(count==1){

                 newStr+=" #unique ";

            }

             newStr+=words[i]+" ";

          }

          System.out.println("Add #Unique:- "+newStr);

    }

    public static void main(String[] args){

        String str = "madam i an a honest man honest means i an the honest ";

        CountWord(str);

        countPalindrome(str);

        CountAll(str);

        ReverseWord(str);

        ReverseAndReplaceArticle(str);

        ReverseAndIntroduce10(str);

        ReverseAndIntroduce1Unique(str);

    }

}

**4.  Assignment**

1. Write a program to reverse first half of integer array.
2. Write a program to compare two String arrays which returns locations of matched items.       (e.g. if String hello is a match  output : String hello Array 1 position x Array 2 position y)
3. Write a program to rotate array by n numbers                                                                                           n should be program argument                                                                                                               e.g.[1,2,3,4]                                                                                                                                           rotation by 1 output [4,1,2,3]
4. Write a program to sort integer array
5. Write a program to find duplicate numbers in array and their frequency.                                      e.g. [1,2,1,3,1,4,2]                                                                                                                                         output: number 1   frequency 3                                                                                                                                 number 2   frequency 2

public class array {

    static void reverseHalfArray(int arr[],int n){

*for*(int i=0;i<n/2-1;i++){

            int temp=arr[i];

            arr[i]=arr[n/2-i-1];

            arr[n/2-i-1]=temp;

         }

         System.out.print("Reverse Half Array:- ");

*for*(int i=0;i<n;i++){

            System.out.print(arr[i]+" ");

         }

    }

    static void compareTwoArray(String s1[],String s2[],int n){

      System.out.print("\nCompare Two String Array:- ");

*for*(int i=0;i<n;i++){

*for*(int j=0;j<n;j++){

*if*(s1[i]==s2[j]){

               System.out.print("\n"+s1[i]+": Array1 Position "+i+"\n"+s2[j]+": Array2 Position "+j);

            }

         }

         }

         System.out.println();

    }

    static void rotateArray(int arr[],int n,int k){

      k=k%n;

*if*(k<0)

      k=k+n;

     reverseArray(arr,0,n-1-k);

     reverseArray(arr,n-k,n-1);

     reverseArray(arr,0,n-1);

*for*(int num*:*arr){

         System.out.print(num+" ");

      }

    }

    static void reverseArray(int[] arr,int start, int end){

*for*(int i=start;i<end;i++){

         int temp=arr[i];

         arr[i]=arr[end];

         arr[end]=temp;

         end--;

      }

    }

    static void sortArray(int arr1[],int n){

      System.out.print("Sorting Array:- ");

*for*(int i=0;i<n;i++){

*for*(int j=i+1;j<n;j++){

*if*(arr1[i]>arr1[j]){

               int temp=0;

               temp=arr1[i];

               arr1[i]=arr1[j];

               arr1[j]=temp;

            }

         }

         System.out.print(arr1[i]+" ");

      }

      System.out.println("\n");

   }

    static void findDuplicateInArray(int arr[],int n){

*for*(int i=0;i<n;i++){

         int count=1;

*for*(int j=i+1;j<n;j++){

*if*(arr[i]==arr[j]){

                count++;

            }

         }

*if*(count>1){

            System.out.println("number "+arr[i]+ " frquency "+" " +count);

         }

      }

   }

     public static void main(String[] args){

        int arr[]={1,2,3,4,5,6,7};

        String s1[]={"A","D","C"};

        String s2[]={"A","C","B"};

        int key=2;

        reverseHalfArray(arr,arr.length);

        compareTwoArray(s1,s2,s1.length);

        sortArray(arr,arr.length);

        findDuplicateInArray(arr,arr.length);

        rotateArray(arr,arr.length,key);

     }

}

public class string\_ex{

    static void CountWord(String str){

        System.out.println("count word = " + str.length());

    }

    static void countPalindrome(String str){

        int count=0;

        String words[]=str.split("\\s");

*for* (String i*:*words){

*if*(isPalindrome(i)){

                count++;

            }

        }

        System.out.println(count);

    }

    static boolean isPalindrome(String i){

        int left=0;

        int right=i.length()-1;

*while*(left<right){

*if*(i.charAt(left)!=i.charAt(right)){

*return* false;

            }

*else*{

                left++;right--;

            }

        }

*return* true;

    }

    static void CountAll(String str){

        int v = 0,c = 0;

*for*(int i=0;i<str.length();i++){

*if*(str.charAt(i)=='a' || str.charAt(i)=='e' || str.charAt(i)=='i' || str.charAt(i)=='o' ||str.charAt(i)=='u' ){v++;}

*else*{c++;}

        }

        String[] s=str.split("//s");

        int a=0,an=0,the=0;

*for* (String word*:*s){

*if*(word.equalsIgnoreCase("a")){

                a++;

            }

*else* *if*(word.equalsIgnoreCase("an")){

                an++;

            }

*else* *if*(word.equalsIgnoreCase("the")){

                the++;

            }

        }

        System.out.println("Vowels = " + v + " Consonents = " + c);

        System.out.println("number of a's = " + a + " number of an's = " + an + " number of the's = " + the +" ");

    }

    static void ReverseWord(String str){

        String words[]=str.split("\\s");

        String reverseWord="";

*for*(String w*:*words){

            StringBuilder sb=*new* StringBuilder(w);

            sb.reverse();

            reverseWord+=sb.toString()+" ";

        }

        System.out.println("Reverse Word = "+reverseWord.trim());

    }

    static String reverse(String str){

        String ReverseStr="";

*for*(int i=str.length()-1;i>=0;i--){

            ReverseStr+=str.charAt(i);

        }

*return* ReverseStr;

    }

    static void ReverseAndReplaceArticle(String str){

        String words[]=str.split("\\s");

        String new\_str="";

*for*(String i*:*words){

*if*(!(i.equals("a") || i.equals("an")|| i.equals("the"))){

                new\_str+=i+" ";

            }

        }

        System.out.println("Remove Article while Reversing:- " + reverse(new\_str));

    }

    static void ReverseAndIntroduce10(String str){

         String words[]=str.split("\\s");

         String d = "";

*for*(int i=0;i<words.length;i++){

*if*(i%10==0){

                d =d+ "01# ";

            }

           d=d+words[i] +" ";

         }

         System.out.println("Add #10:- "+reverse(d));

    }

    static void ReverseAndIntroduce1Unique(String str){

         String words[]=str.split("\\s");

         String newStr="";

*for*(int i=0;i<words.length;i++){

            int count=1;

*for*(int j=i+1;j<words.length;j++){

*if*(words[i].equalsIgnoreCase(words[j])){

                    count++;

                }

            }

*if*(count==1){

                 newStr+=" #unique ";

            }

             newStr+=words[i]+" ";

          }

          System.out.println("Add #Unique:- "+newStr);

    }

    public static void main(String[] args){

        String str = "madam i an a honest man honest means i an the honest ";

        CountWord(str);

        countPalindrome(str);

        CountAll(str);

        ReverseWord(str);

        ReverseAndReplaceArticle(str);

        ReverseAndIntroduce10(str);

        ReverseAndIntroduce1Unique(str);

    }

}