

Q1. Explain Encapsulation in java programming?

Ans:

Hiding non-essentials information to user.

You achieve Encapsulation using

1. Declare member data as private
2. Access private variable using public method getter.
3. Assign /change value of private member data using setter

Properties:(getter):

1. The Return type of getter method is always same as Its member data , data type.
2. Getter method does not take any argument
3. Name of the getter name

Example:

Prefix+suffix()

get+memberData(){

}

```
class Student{
```

```
    private String name="Ram";
```

```
    private String enroll="0103CS";
```

```
    public String getName(){
```

```
        return name;
```

```
}
```

```
    public String getEnroll(){  
        return enroll;  
    }  
}
```

2. Setter:

A. The Return type of setter method is always void

B. Setter method is take one argument which is same its member data.

C. Syntax

Prefix+suffix(Data type member data){

}

```
class Student{  
    private String name="Ram";  
    private String enroll="0103CS";  
  
    public String getName(){  
        return name;  
    }  
  
    public String getEnroll(){  
        return enroll;  
    }  
}
```

```
public void setName(String name){
this.name=name;
}

public void setEnroll(String enroll){
this.enroll=enroll;
}
}

class Test{

public static void main(String args[]){
Student s=new Student();
System.out.println("Name : "+s.getName());
System.out.println("Enroll : "+s.getEnroll());
s.setName("AKASH");
s.setEnroll("0131CS");
System.out.println("Name : "+s.getName());
System.out.println("Enroll : "+s.getEnroll());

}
}
```

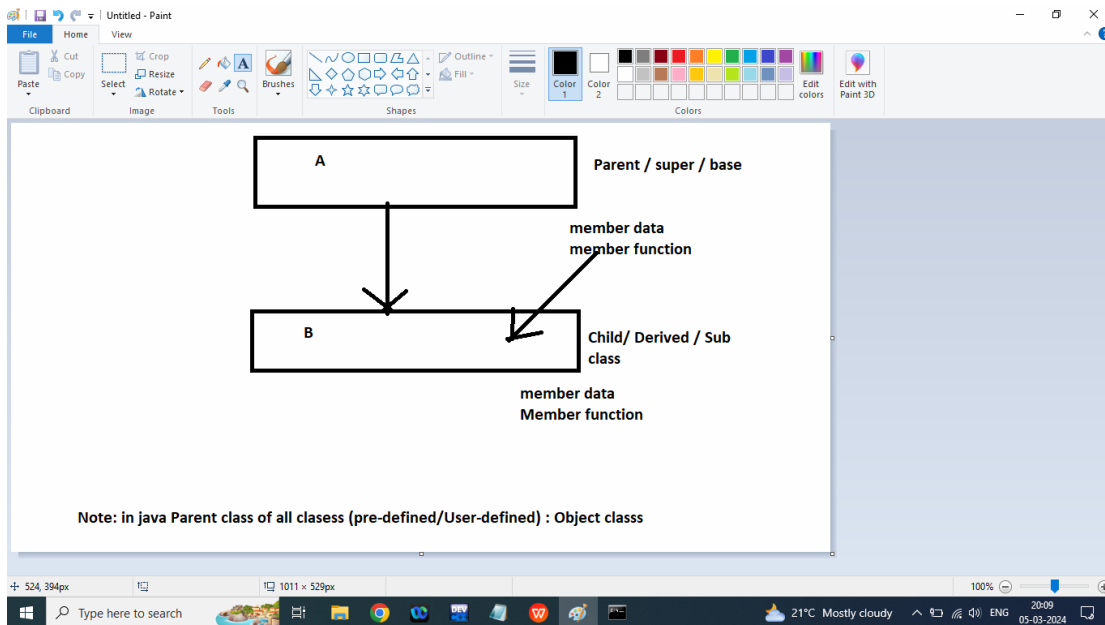
Q2.Explain Inheritance in java programming?

Ans:

Passing properties from one class to another class is known as Inheritance

Properties may be member data or member function

In case of Inheritance member data or member function should not be private.



```
class Parent{
//member data
//member function
}

class Child extends Parent{
//Member data
//member function
}
```

```
class Parent{
    String fname="KARTIK";
    String mname="shivani";
    public void show(){
        System.out.println(fname+" : "+mname);
    }
    public static void main(String args[]){
        System.out.println("This is Parent class Main Method");
    }
}
```

```

    }
}

class Child extends Parent{
public static void main(String args[]){
    System.out.println("This is Child class Main Method");
}
}

```

Q2. Explain MergeSort in data Structure?

Ans:

The screenshot shows a Microsoft Paint window titled "Untitled - Paint". The interface includes a ribbon with tabs for File, Home, View, and Text. The Text tab is active, showing options for font (Calibri, size 24), bold, italic, underline, and background color (Opaque, Transparent). A color palette is visible on the right.

In the center of the canvas, there is a diagram of an array with 8 elements: 10, 50, 20, 80, 30, 70, 40, and 60. Below the array, indices 0 through 7 are listed. A vertical blue line is drawn between index 3 and index 4, labeled "mid" in red. Below index 0 is the label "si" in red, and below index 7 is the label "ei" in red.

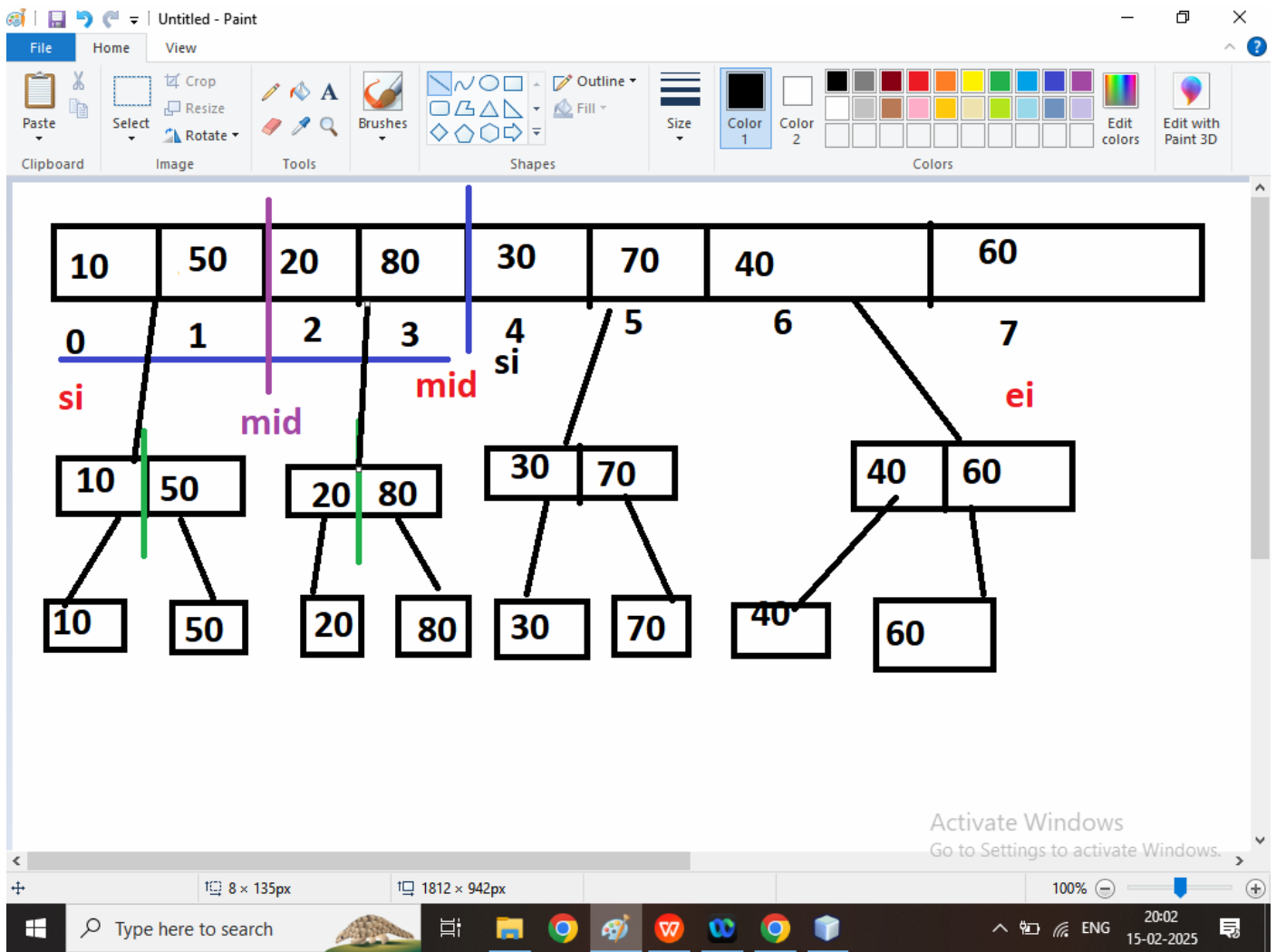
Below the array diagram, there is a code snippet enclosed in a dashed blue box:

```

if(si >= ei){
    return;
}
divide(arr, si, mid) // left array
divide(arr, mid+1, ei) // right array
conquer()

```

At the bottom of the window, there is a status bar showing dimensions (693, 328px), zoom level (100%), and a taskbar with various application icons. A Windows watermark "Activate Windows Go to Settings to activate Windows." is visible in the bottom right corner.



Microsoft Paint interface showing a diagram of a binary search tree construction process. The diagram illustrates the recursive steps for inserting elements into a binary tree.

Initial Array: 10, 50, 20, 80, 30, 70, 40, 60

Indices: 0, 1, 2, 3, 4, 5, 6, 7

Labels: si (start index), mid (middle index), ei (end index)

Tree Structure:

- Root Node: 50 (Index 1)
 - Left Child: 10 (Index 0)
 - Left Child: 10 (Index 0)
 - Right Child: 50 (Index 1)
 - Right Child: 20 (Index 2)
 - Left Child: 20 (Index 2)
 - Right Child: 80 (Index 3)
- Right Child: 30 (Index 4)
 - Left Child: 30 (Index 4)
 - Right Child: 70 (Index 5)
- Right Child: 40 (Index 6)
 - Left Child: 40 (Index 6)
 - Right Child: 60 (Index 7)

The diagram shows the recursive process of inserting elements into a binary tree. The root node is 50. Its left child is 10, and its right child is 20. The left child of 10 is 10, and the right child of 10 is 50. The left child of 20 is 20, and the right child of 20 is 80. The right child of 50 is 30, and the right child of 30 is 70. The right child of 50 is 40, and the right child of 40 is 60.

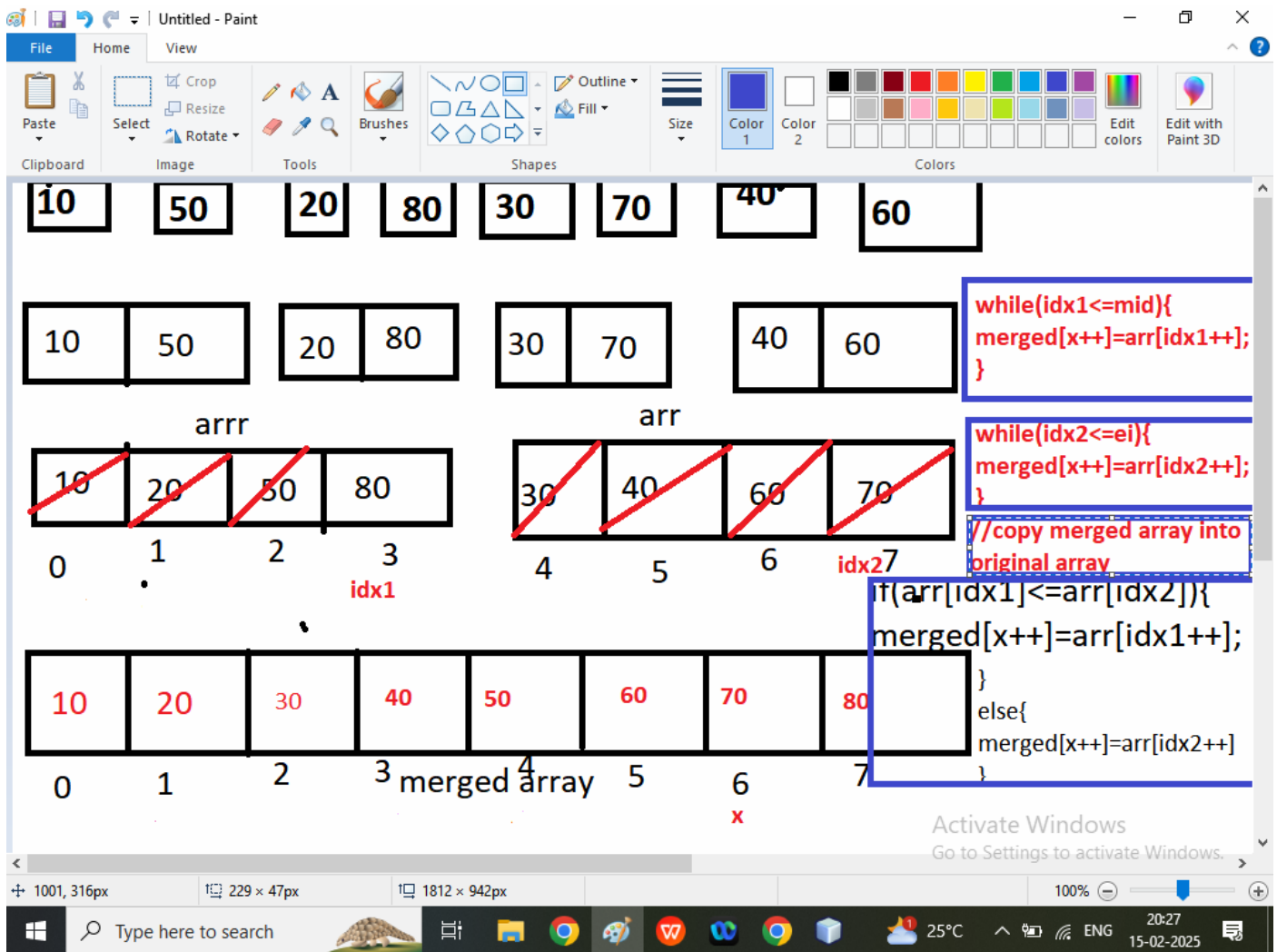
Microsoft Paint window titled "Untitled - Paint". The interface includes a ribbon with tabs for File, Home, and View. The Home tab is active, showing options for Clipboard (Paste), Image (Select, Crop, Resize, Rotate), Tools (Brushes, Pencil, Eraser, Highlighter, Ink, Lasso, Magnifying Glass), Shapes (various geometric shapes), Outline (Fill, Stroke), Size, Color (Color 1, Color 2), Colors (a grid of color swatches), Edit colors, and Edit with Paint 3D.

The main canvas displays a diagram illustrating the merging of two arrays. At the top, a sequence of numbers is shown in individual boxes: 10, 50, 20, 80, 30, 70, 40, 60. Below this, two arrays are defined:

- arr1**: A horizontal array of four boxes containing the values 10, 20, 50, and 80. Below the boxes are indices 0, 1, 2, and 3. A red label **idx1** is positioned below the index 0.
- arr2**: A horizontal array of four boxes containing the values 30, 40, 60, and 70. Below the boxes are indices 4, 5, 6, and 7. A red label **idx2** is positioned below the index 4.

Below these two arrays is a single horizontal array of eight empty boxes, labeled **merged array** underneath. The diagram is intended to show the step-by-step merging of the two input arrays into the merged array.

At the bottom of the window, there is a status bar showing dimensions (675, 451px), zoom level (100%), and a Windows taskbar with various application icons and system information (25°C, 20:11, 15-02-2025).



Descending Order

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dsafeb2025;

```

```
import javafx.beans.binding.Bindings;
```

```

public class MergeSortDemo {
    public static void conquer(int arr[],int si,int mid,int ei)
    {
        int size=ei-si+1;
        int merged[]=new int[size];
        int idx1=si;

        int idx2=mid+1;
        int x=0;//x represent merged array
        while(idx1<=mid && idx2<=ei){
            if(arr[idx1]>=arr[idx2]){
                merged[x++]=arr[idx1++];
            }else{
                merged[x++]=arr[idx2++];
            }
        }
        while(idx1<=mid){

```

```

        merged[x++]=arr[idx1++];
    }
    while(idx2<=ei){
        merged[x++]=arr[idx2++];
    }

    //copy the data into original array
    for(int i1=0,j1=si;i1<size;i1++,j1++){
        arr[j1]=merged[i1];
    }
}

public static void divide(int arr[],int si,int ei){
    if(si>=ei){//base condition
        return;
    }
    int mid=(si+ei)/2;
    divide(arr, si, mid);//for left sub array
    divide(arr, mid+1, ei);//for right sub array
    conquer(arr,si,mid,ei);//
}

public static void main(String[] args) {
    int arr[]={10,30,20,50,40,5};
    System.out.println("Print Before Sorting ");
    for(int i=0;i<arr.length;i++){
        System.out.print("\t"+arr[i]);
    }
    divide(arr, 0, arr.length-1);
    System.out.println("\nPrint After Sorting ");
    for(int i=0;i<arr.length;i++){
        System.out.print("\t"+arr[i]);
    }
}
}

```

Ascending Order

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package dsafeb2025;

import javafx.beans.binding.Bindings;

```

```

public class MergeSortDemo {
    public static void conquer(int arr[],int si,int mid,int ei)
    {
        int size=ei-si+1;
        int merged[]=new int[size];
        int idx1=si;

        int idx2=mid+1;
        int x=0;//x represent merged array
        while(idx1<=mid && idx2<=ei){
            if(arr[idx1]<=arr[idx2]){
                merged[x++]=arr[idx1++];
            }else{

```

```

        merged[x++]=arr[idx2++];
    }
}
while(idx1<=mid){
    merged[x++]=arr[idx1++];
}
while(idx2<=ei){
    merged[x++]=arr[idx2++];
}

//copy the data into original array
for(int i1=0,j1=si;i1<size;i1++,j1++){
    arr[j1]=merged[i1];
}
}

public static void divide(int arr[],int si,int ei){
    if(si>=ei){//base condition
        return;
    }
    int mid=(si+ei)/2;
    divide(arr, si, mid);//for left sub array
    divide(arr, mid+1, ei);//for right sub array
    conquer(arr,si,mid,ei);//
}

public static void main(String[] args) {
    int arr[]={10,30,20,50,40,5};
    System.out.println("Print Before Sorting ");
    for(int i=0;i<arr.length;i++){
        System.out.print("\t"+arr[i]);
    }
    divide(arr, 0, arr.length-1);
    System.out.println("\nPrint After Sorting ");
    for(int i=0;i<arr.length;i++){
        System.out.print("\t"+arr[i]);
    }
}
}

```

Untitled - Paint

File Home View

Paste Select Crop Resize Rotate Image Tools Brushes Shapes Outline Fill Size Color 1 Color 2 Colors Edit colors Edit with Paint 3D

10 50 20 80 30 70 40 60

10 50 20 80 30 70 40 60

arr arr

10 20 50 80 30 40 60 70

0 1 2 3 4 5 6 7

idx1 idx idx2

if(arr[idx1]<=arr[idx2]){
merged[x++]=arr[idx1++];
}

merged array

Activate Windows
Go to Settings to activate Windows.

39,321px 61 x 1px 1812 x 942px 100%

Windows Type here to search 25°C 20:13 15-02-2025

Untitled - Paint

File Home View

Paste Select Crop Resize Rotate Image Tools Brushes Shapes Outline Fill Size Color 1 Color 2 Colors Edit colors Edit with Paint 3D

10 50 20 80 30 70 40 60

10 50 20 80 30 70 40 60

arrr arr

10 20 50 80 30 40 60 70

0 1 2 3 4 5 6 7

idx idx2

if(arr[idx1]<=arr[idx2]){
merged[x++]=arr[idx1++];
}

10 merged array 4

0 1 2 3 4 5 6 7

x

Activate Windows
Go to Settings to activate Windows.

1812 x 942px 100%

Type here to search 25°C 20:14 15-02-2025

Untitled - Paint

File Home View

Paste Select Crop Resize Rotate Image Tools Brushes Shapes Outline Fill Size Color 1 Color 2 Colors Edit colors Edit with Paint 3D

10 50 20 80 30 70 40 60

10 50 20 80 30 70 40 60

arrr arr

10 20 50 80 30 40 60 70

0 1 2 3 4 5 6 7

idx1 idx2

if(arr[idx1]<=arr[idx2]){
merged[x++]=arr[idx1++];
}

10 20 merged array 4 5 6 7

0 1 2 3 4 5 6 7

Activate Windows
Go to Settings to activate Windows.

+ 155,359px 1812 x 942px 100%

Windows Type here to search 25°C 20:15 15-02-2025

Untitled - Paint

File Home View

Paste Select Crop Resize Rotate Image Tools Brushes Shapes Outline Fill Size Color 1 Color 2 Colors Edit colors Edit with Paint 3D

10 50 20 80 30 70 40 60

10 50 20 80 30 70 40 60

arrr arr

10 20 50 80 30 40 60 70

0 1 2 3 4 5 6 7

idx1 idx2

if(arr[idx1] <= arr[idx2]){
merged[x++] = arr[idx1++];
}
else{
merged[x++] = arr[idx2++];
}

10 20 30 merged array 4 5 6 7

0 1 2 3 4 5 6 7

x

Activate Windows
Go to Settings to activate Windows.

+ 932, 206px 69 x 75px 1812 x 942px 100%

Windows Type here to search 25°C 20:17 15-02-2025

Untitled - Paint

File Home View

Paste Select Crop Resize Rotate Image Tools Brushes Shapes Outline Fill Size Color 1 Color 2 Colors Edit colors Edit with Paint 3D

10 50 20 80 30 70 40 60

10 50 20 80 30 70 40 60

arr

10 20 50 80

0 1 2 3

idx1

arr

30 40 60 70

4 5 6 7

idx2

```
while(idx1<=mid){
merged[x++]=arr[idx1++];
}

while(idx2<=ei){
merged[x++]=arr[idx2++];
}

//copy merged array into
original array
if(arr[idx1]<=arr[idx2]){
merged[x++]=arr[idx1++];
}
else{
merged[x++]=arr[idx2++];
}
```

merged array

10 20 30 40 50 60 70 80

0 1 2 3 4 5 6 7

x

Activate Windows
Go to Settings to activate Windows.

674, 191px 1812 x 942px 100%

Type here to search 25°C 20:23 15-02-2025