

Arrays & Strings

Stores data elements based on an sequential, most commonly 0 based, index.

Time Complexity

- **Indexing:** Linear array: O(1), Dynamic array: O(1)
- **Search:** Linear array: O(n), Dynamic array: O(n)
- **Optimized Search:** Linear array: O(log n), Dynamic array: O(log n)
- **Insertion:** Linear array: n/a, Dynamic array: O(n)

Bonus:

- type[] name = {val1, val2, ...}
- Arrays.sort(arr) -> O(n log(n))
- Collections.sort(list) -> O(n log(n))
- int digit = '4' - '0' -> 4
- String s = String.valueOf('e') -> "e"
- (int) 'a' -> 97 (ASCII)
- new String(char[] arr) ['a','e'] -> "ae"
- (char) ('a' + 1) -> 'b'
- Character.isLetterOrDigit(char) -> true/false
- new ArrayList<>(anotherList); -> list w/ items
- StringBuilder.append(char||String)

Stack/Queue/Deque

Stack	Queue	Deque	Heap
Last In First Out	First In Last Out	Provides first/last	Ascending Order
push(val)	offer(val)	offer(val)	offer(val)
pop()	poll()	poll()	poll()
peek()	peek()	peek()	peek()

Implementation in Java:

- Stack<E> stack = new Stack();
- Queue<E> queue = new LinkedList();
- Deque<E> deque = new LinkedList();
- PriorityQueue<E> pq = new PriorityQueue();

DFS & BFS Big O Notation

	Time	Space
DFS	O(E+V)	O(Height)
BFS	O(E+V)	O(Length)

V & E -> where V is the number of vertices and E is the number of edges.

Height -> where h is the maximum height of the tree.

Length -> where l is the maximum number of nodes in a single level.

DFS vs BFS

DFS	BFS
• Better when target is closer to Source.	• Better when target is far from Source.
• Stack -> LIFO	• Queue -> FIFO
• Preorder, Inorder, Postorder Search	• Level Order Search
• Goes deep	• Goes wide
• Recursive	• Iterative
• Fast	• Slow

Linked List

Stores data with nodes that point to other nodes.

Time Complexity

- **Indexing:** O(n)
- **Search:** O(n)
- **Optimized Search:** O(n)
- **Append:** O(1)
- **Prepend:** O(1)
- **Insertion:** O(n)

HashTable

Stores data with key-value pairs.

Time Complexity

- **Indexing:** O(1)
- **Search:** O(1)
- **Insertion:** O(1)

Bonus:

- {1, -1, 0, 2, -2} into map
- HashMap {-1, 0, 2, 1, -2} -> any order
- LinkedHashMap {1, -1, 0, 2, -2} -> insertion order
- TreeMap {-2, -1, 0, 1, 2} -> sorted
- Set doesn't allow duplicates.
- map.getOrDefault(key, default value)



BFS Impl for Graph

```
public boolean connected(int[][] graph, int start, int end) {  
    Set<Integer> visited = new HashSet<>();  
    Queue<Integer> toVisit = new LinkedList<>();  
    toVisit.enqueue(start);  
    while (!toVisit.isEmpty()) {  
        int curr = toVisit.dequeue();  
        if (visited.contains(curr)) continue;  
        if (curr == end) return true;  
        for (int i : graph[start]) {  
            toVisit.enqueue(i);  
        }  
        visited.add(curr);  
    }  
    return false;  
}
```

BFS Impl. for Level-order Tree Traversal

```
private void printLevelOrder(TreeNode root) {  
    Queue<TreeNode> queue = new LinkedList<>();  
    queue.offer(root);  
    while (!queue.isEmpty()) {  
        TreeNode tempNode = queue.poll();  
        print(tempNode.data + " ");  
  
        //add left child  
        if (tempNode.left != null) {  
            queue.offer(tempNode.left);  
        }  
  
        //add right right child  
        if (tempNode.right != null) {  
            queue.offer(tempNode.right);  
        }  
    }  
}
```

DFS Impl for Graph

```
public boolean connected(int[][] graph, int start, int end) {  
    Set<Integer> visited = new HashSet<>();  
    return connected(graph, start, end, visited);  
}  
  
private boolean connected(int[][] graph, int start, int end, Set<Integer> visited) {  
    if (start == end) return true;  
    if (visited.contains(start)) return false;  
    visited.add(start);  
    for (int i : graph[start]) {  
        if (connected(graph, i, end, visited)) {  
            return true;  
        }  
    }  
    return false;  
}
```

DFS Impl. for In-order Tree Traversal

```
private void inorder(TreeNode TreeNode) {  
    if (TreeNode == null)  
        return;  
    // Traverse left  
    inorder(TreeNode.left);  
    // Traverse root  
    print(TreeNode.data + " ");  
    // Traverse right  
    inorder(TreeNode.right);  
}
```



By burcuco

cheatography.com/burcuco/

Published 30th March, 2021.

Last updated 30th March, 2021.

Page 2 of 7.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Dynamic Programming

- Dynamic programming is the technique of storing repeated computations in memory, rather than recomputing them every time you need them.
- The ultimate goal of this process is to improve runtime.
- Dynamic programming allows you to use more space to take less time.

Dynamic Programming Patterns

- Minimum (Maximum) Path to Reach a Target

Approach:

Choose minimum (maximum) path among all possible paths before the current state, then add value for the current state.

Formula:

$\text{routes}[i] = \min(\text{routes}[i-1], \text{routes}[i-2], \dots, \text{routes}[i-k]) + \text{cost}[i]$

- Distinct Ways

Approach:

Choose minimum (maximum) path among all possible paths before the current state, then add value for the current state.

Formula:

$\text{routes}[i] = \text{routes}[i-1] + \text{routes}[i-2], \dots, + \text{routes}[i-k]$

- Merging Intervals

Approach:

Find all optimal solutions for every interval and return the best possible answer

Formula:

$\text{dp}[i][j] = \text{dp}[i][k] + \text{result}[k] + \text{dp}[k+1][j]$

- DP on Strings

Approach:

Compare 2 chars of String or 2 Strings. Do whatever you do. Return.

Formula:

if $s1[i-1] == s2[j-1]$ then $\text{dp}[i][j] = //\text{code}$.

Else $\text{dp}[i][j] = //\text{code}$

- Decision Making

Approach:

If you decide to choose the current value use the previous result where the value was ignored; vice-versa, if you decide to ignore the current value use previous result where value was used.

Formula:

$\text{dp}[i][j] = \max(\{\text{dp}[i][j], \text{dp}[i-1][j] + \text{arr}[i], \text{dp}[i-1][j-1]\})$;

$\text{dp}[i][j-1] = \max(\{\text{dp}[i][j-1], \text{dp}[i-1][j-1] + \text{arr}[i], \text{arr}[i]\})$;

Binary Search Big O Notation

	Time	Space
Binary Search	$O(\log n)$	$O(1)$

Binary Search - Recursive

```
public int binarySearch(int search, int[] array, int start, int end) {  
    int middle = start + ((end - start) / 2);  
    if(end < start) {  
        return -1;  
    }  
    if (search == array[middle]) {  
        return middle;  
    } else if (search < array[middle]) {  
        return binarySearch(search, array, start, middle - 1);  
    } else {  
        return binarySearch(search, array, middle + 1, end);  
    }  
}
```

Binary Search - Iterative

```
public int binarySearch(int target, int[] array) {  
    int start = 0;  
    int end = array.length - 1;  
    while (start <= end) {  
        int middle = start + ((end - start) / 2);  
        if (target == array[middle]) {  
            return target;  
        } else if (target < array[middle]) {  
            end = middle - 1;  
        } else {  
            start = middle + 1;  
        }  
    }  
}
```



Cheatography

Data Structures and Algorithms Cheat Sheet by burcuco via cheatography.com/133629/cs/27343/

Binary Search - Iterative (cont)

```
    return -1;  
}
```

Bit Manipulation

Sign Bit	0 -> Positive, 1 -> Negative
----------	------------------------------

AND	0 & 0 -> 0
	0 & 1 -> 0
	1 & 1 -> 1

OR	0 0 -> 0
	0 1 -> 1
	1 1 -> 1

XOR	0 ^ 0 -> 0
	0 ^ 1 -> 1
	1 ^ 1 -> 0

INVERT	~ 0 -> 1
	~ 1 -> 0

Bonus:

- Shifting

- Left Shift

- 0001 << 0010 (Multiply by 2)

- Right Shift

- 0010 >> 0001 (Division by 2)

- Count 1's of n, Remove last bit

- n = n & (n-1);

- Extract last bit

- n&-n or n&~(n-1) or n^(n&(n-1))

- n ^ n -> 0

- n ^ 0 -> n

Sorting Big O Notation

	Best	Average	Space
Merge Sort	O(n log(n))	O(n log(n))	O(n)
Heap Sort	O(n log(n))	O(n log(n))	O(1)
Quick Sort	O(n log(n))	O(n log(n))	O(log(n))
Insertion Sort	O(n)	O(n^2)	O(1)
Selection Sort	O(n^2)	O(n^2)	O(1)
Bubble Sort	O(n)	O(n^2)	O(1)

Merge Sort

```
private void mergesort(int low, int high) {  
    if (low < high) {  
        int middle = low + (high - low) / 2;  
        mergesort(low, middle);  
        mergesort(middle + 1, high);  
        merge(low, middle, high);  
    }  
}  
  
private void merge(int low, int middle, int high)  
{  
    for (int i = low; i <= high; i++) {  
        helper[i] = numbers[i];  
    }  
    int i = low;  
    int j = middle + 1;  
    int k = low;  
    while (i <= middle && j <= high) {  
        if (helper[i] <= helper[j]) {  
            numbers[k] = helper[i];  
            i++;  
        } else {  
            numbers[k] = helper[j];  
            j++;  
        }  
        k++;  
    }  
    while (i <= middle) {  
        numbers[k] = helper[i];  
        k++;  
        i++;  
    }  
}
```

Quick Sort

```
private void quicksort(int low, int high) {  
    int i = low, j = high;  
    int pivot = numbers[low + (high-low)/2];  
    while (i <= j) {  
        while (numbers[i] < pivot) {  
            i++;  
        }  
        while (numbers[j] > pivot) {  
            j--;  
        }  
        if (i <= j) {  
            exchange(i, j);  
            i++;  
            j--;  
        }  
    }  
    if (low < j)  
        quicksort(low, j);  
    if (i < high)  
        quicksort(i, high);  
}
```



By **burcuco**
cheatography.com/burcuco/

Published 30th March, 2021.
Last updated 30th March, 2021.
Page 4 of 7.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Insertion Sort

```
void insertionSort(int arr[]) {  
    int n = arr.length;  
    for (int i = 1; i < n; ++i) {  
        int key = arr[i];  
        int j = i - 1;  
        while (j >= 0 && arr[j] > key) {  
            arr[j + 1] = arr[j];  
            j = j - 1;  
        }  
        arr[j + 1] = key;  
    }  
}
```

Combinations Backtrack Pattern (cont)

```
    }  
}  
}
```

Palindrome Backtrack Pattern

- Palindrome Partitioning

```
public List<List<String>> partition(String s) {  
    List<List<String>> list = new ArrayList<>();  
    backtrack(list, new ArrayList<>(), s, 0);  
    return list;  
}  
  
public void backtrack(List<List<String>> list,  
List<String> tempList, String s, int start){  
    if(start == s.length())  
        list.add(new ArrayList<>(tempList));  
    else{  
        for(int i = start; i < s.length(); i++){  
            if(isPalindrome(s, start, i)){  
                tempList.add(s.substring(start, i +  
1));  
                backtrack(list, tempList, s, i + 1);  
                tempList.remove(tempList.size() - 1);  
            }  
        }  
    }  
}
```

Subsets Backtrack Pattern

- Subsets

```
public List<List<Integer>> subsets(int[] nums) {  
    List<List<Integer>> list = new ArrayList<>();  
    Arrays.sort(nums);  
    backtrack(list, new ArrayList<>(), nums, 0);  
    return list;
```

Combinations Backtrack Pattern

- Combination

```
public List<List<Integer>> combinationSum(int []  
nums, int target) {  
    List<List<Integer>> list = new ArrayList<>();  
    Arrays.sort(nums);  
    backtrack(list, new ArrayList<>(), nums,  
target, 0);  
    return list;  
}  
  
private void backtrack(List<List<Integer>> list,  
List<Integer> tempList, int [] nums, int remain,  
int start){  
    if(remain < 0) return;  
    else if(remain == 0) list.add(new ArrayList<>  
(tempList));  
    else{  
        for(int i = start; i < nums.length; i++){  
            tempList.add(nums[i]);  
            // not i + 1 because we can reuse  
            same elements  
            backtrack(list, tempList, nums, remain  
- nums[i], i);  
            // not i + 1 because we can reuse  
            same elements  
            tempList.remove(tempList.size() - 1);  
    }
```



Subsets Backtrack Pattern (cont)

```
}

private void backtrack(List<List<Integer>> list,
List<Integer> tempList, int [] nums, int start){
    list.add(new ArrayList<>(tempList));
    for(int i = start; i < nums.length; i++){
        // skip duplicates
        if(i > start && nums[i] == nums[i-1])
            continue;
        // skip duplicates
        tempList.add(nums[i]);
        backtrack(list, tempList, nums, i + 1);
        tempList.remove(tempList.size() - 1);
    }
}
```

Permutations Backtrack Pattern (cont)

```
}

}
```

Permutations Backtrack Pattern

- Permutations

```
public List<List<Integer>> permute(int[] nums) {
    List<List<Integer>> list = new ArrayList<>();
    // Arrays.sort(nums); // not necessary
    backtrack(list, new ArrayList<>(), nums);
    return list;
}

private void backtrack(List<List<Integer>> list,
List<Integer> tempList, int [] nums){
    if(tempList.size() == nums.length){
        list.add(new ArrayList<>(tempList));
    } else{
        for(int i = 0; i < nums.length; i++){
            // element already exists, skip
            if(tempList.contains(nums[i])) continue;
            // element already exists, skip
            tempList.add(nums[i]);
            backtrack(list, tempList, nums);
            tempList.remove(tempList.size() - 1);
        }
    }
}
```



By burcuco

cheatography.com/burcuco/

Published 30th March, 2021.

Last updated 30th March, 2021.

Page 6 of 7.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Code Analysis – Sample Questions

Because the event rules may vary from year to year, these sample questions may address topics not included in the current year's rules.

1. What is the printed output of the following program?

```
// Example1.java

public class Example1
{
    public void one ()
    {
        System.out.println("three");
    }

    public void two ()
    {
        System.out.println("two");
    }

    public void three ()
    {
        one();
    }

    public static void main (String[] args)
    {
        three();
        two();
        one();
    }
}
```

Write your answer below.

/ 15 points

Solution:

three

two

three

Incorrect solutions:

threetwothree

The printing is not on separate lines. 10 points.

one

two

three

The first line is incorrect. 10 points for the last two correct lines.

two

three

three

Even though all three lines are present, they are in the wrong order. Only the last line counts, 5 points.

Discussion:

This question tests basic knowledge of method calls and printing. It is a simple enough question that all teams should be able to answer (a good exam should have one such question). It also demonstrates one possible way to score the problem. Because the number and order of output lines matter, horizontal writing lines are provided.

2. What is the printed output of the following program?

```
private int a = 5;  
private int b = 10;  
  
public static void main (String[] args)  
{  
    int result = a + b * 4;  
    System.out.println(result + 3);  
}
```

Write your answer below.

/ 20 points

Solution:

48

Incorrect solutions:

63

This would be the result if expressions were evaluated left-to-right; however, multiplication takes precedence over addition. 5 points.

453

The output should be mathematically added, not concatenated. 5 points.

Discussion:

This question tests integer variables, integer operators, and precedence.

3. What is the printed output of the following program?

```
public static void main (String[] args)
{
    System.out.println(4%8, 14%9, 20%7);
}
```

Write your answer below.

/ 25 points

Solution:

456

(The last 2 lines should be blank.)

Incorrect solutions:

4

5

6

The printing should not be on separate lines. 15 points.

Discussion:

This question tests the modulus operator and the concatenation operation of `System.out.println`. It also demonstrates that it is not necessary to use every line of the answer space.

4. What is the printed output of the following code fragment?

```
String s = "This is a test.";
int i = 0;
while (i < 10) {
    char c = s[i];
    if (c >= 'a' && c <= 'z')
        c = c - 'a' + 'A';
    else if (c >= 'A' && c <= 'Z')
        c = c - 'A' + 'a';
    System.out.print(c);
    i = i + 1;
}
```

Write your answer below.

/ 25 points

Solution:

tHIS 1S A

Incorrect solutions:

tHIS 1S A TEST.

The loop executes for only 10 characters, not the full length of the string. 20 points.

tHIS iS A

The '\$' and '1' characters are outside the ranges that are tested, and therefore remain unchanged. 20 points.

Discussion:

This question tests knowledge of loops, strings, and character arithmetic. The loop is kept simple for beginning programmers (for example, using `i = i+1` rather than `i++`). Possible variants could include dropping the else, and replacing `>=` with `>`.

5. What is the printed output of the following code fragment?

```
int x = 5;
boolean done = false;
while (!done) {
    System.out.print(x);
    done = (--x < 1);
}
System.out.println(" liftoff!");
```

Write your answer below.

/ 25 points

Solution:

54321 liftoff!

Incorrect solutions:

543210 liftoff!

The loop should terminate as soon as x becomes 0. 20 points.

Discussion:

This question tests boolean variables, the logical not operator, while loops, and predecrement. Possible variants could include starting with `done = true`, or using a postdecrement.

6. What is the printed output of the following code fragment?

```
int[] fibs = { 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 };  
int q = 10;  
int lo = 0; hi = 8;  
while (lo < hi) {  
    int mid = (lo+hi) / 2;  
    if (q < fibs[mid])  
        hi = mid;  
    else  
        lo = mid;  
}  
System.out.println(lo);
```

Write your answer below.

/ 40 points

Solution:

5

Incorrect solutions:

6

When the searched value is between two array values, the program should print the *lower* index.
30 points.

8

Prints the index, not the array value. 30 points.

Discussion:

This is a binary search. It also tests knowledge of arrays.

7. What is the printed output of the following code fragment?

```
String a = "SCIENCE OLYMPIAD";
String b = a.charAt(a.length - 2);
String c = a.substring(8, 1);
String d = a.substring(3, 2);
String e = d.substring(1, 1).concat(d.substring(0, 1));
System.out.println(b, c, e);
```

Write your answer below.

/ 25 points

Solution:

DONE

5 points for each letter in any order, plus 5 points for the complete correct order.

Discussion:

This question tests knowledge of string operations.

8. What is the printed output of the code below when it is called as `f(12)`?

```
private int primes[] = { 2, 3, 5, 7, 11, 13, 17, 19 };

public void f (int n)
{
    int i = 0;

    while (i < 8) {
        int d = primes[i];
        if ((n % d) == 0) {
            System.out.printf("%d ", d);
            f(n/d);
            return;
        }
        i = i + 1;
    }
    System.out.print("\n");
    return;
}
```

Write your answer below.

/ 25 points

Solution:

2 2 3

Incorrect solutions:

2 3

Each time the method is called, it restarts the factor search. 20 points.

Discussion:

This is an example of a recursive method. It also shows how a problem can be specified as a method call – in this case, f(12).

9. What is the printed output of the following program?

```
// Scope.java

public class Scope
{
    private static int x = 1;

    public static void main(String[] args)
    {
        int x = 2;

        System.out.printf("%d ", x);
        a();
        b();
        a();
        b();
        System.out.printf("%d ", x);
    }

    public static void a()
    {
        int x = 3;

        System.out.printf("%d ", x++);
        System.out.printf("%d ", x);
    }

    public static void b()
    {
        System.out.printf("%d ", x);
        x *= 10;
        System.out.printf("%d ", x);
    }
}
```

Write your answer below.

/ 30 points

Solution:

2 3 4 1 10 3 4 10 100 2
3 points per correct answer.

Discussion:

This question tests variable scope. Because a horizontal answer blank is given, the spacing of the answer does not matter.

10. What is the printed output of the following code fragment?

```
private int array[] = { 15, 4, 16, 3, 9, 11, 1, 12, 19, 6 };

int x = 0;
while (x < 9) {
    int y = x+1;
    while (y < 10) {
        if (array[x] > array[y]) {
            int tmp = array[x];
            array[x] = array[y];
            array[y] = tmp;
        }
        y = y + 1;
    }
    x = x + 1;
}

int i = 0;
while (i < 10) {
    System.out.printf("%d  ", array[i]);
    i = i + 1;
}
```

Write your answer below.

/ 40 points

Solution:

1 3 4 6 9 11 12 15 16 19

3 points per number, plus 10 points for the complete correct order.

Discussion:

This is an example of sorting (a bubble sort).

11. What is the printed output of the following program?

```
public void main(){int p=9,q=1,t=0;while(q<p){t+=q;int n=1;  
while(n<=q)System.out.print('X');System.out.print('\n');q+=  
2;}System.out.printf("%d\n",t);}
```

Write your answer below.

/ 30 points

Solution:

X

XXX

XXXX

XXXXXX

16

[extra lines at the end should be left blank]

5 points for each correct X line, plus 10 points for the correct final total.

Discussion:

This code is "obfuscated". It is intentionally made compact and difficult to read.

12. The following program "paints" and then prints a 10x10 array of characters. Find the printed output.

```
private static char[][] grid = new char[10][10];

public static void initgrid ()
{
    int x = 0;
    while (x < 10) {
        int y = 0;
        while (y < 10)
            grid[x][y++] = '.';
        x++;
    }
    return;
}

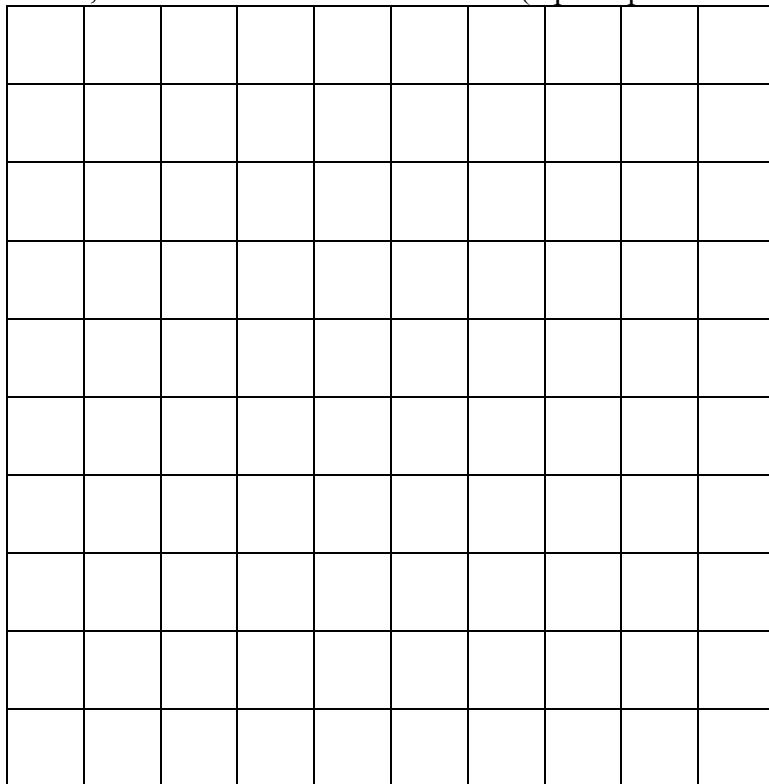
public static void
bresenham (int x0, int y0, int x1, int y1, char ch)
{
    int x = x0, y = y0;
    int dx = x1 - x0, dy = y1 - y0;
    int D = 2*dy - dx;

    // The following code assumes dx > dy > 0.
    do {
        grid[x][y] = ch;
        if (D > 0) {
            D = D - 2*dx;
            y++;
        }
        D = D + 2*dy;
    } while (x++ < x1);
    return;
}

public static void printgrid ()
{
    int y = 0;
    while (y < 10) {
        int x = 0;
        while (x < 10) {
            System.out.print(grid[x][y]);
            x++;
        }
        System.out.print('\n');
        y++;
    }
}
```

```
    return;  
}  
  
public static void main ()  
{  
    initgrid();  
  
    bresenham(1,1, 8,6, '#');  
  
    printgrid();  
}
```

Draw your answer below, with one character in each cell. (1 point per character) / 100 points



Solution:

.
.	#
.	.	#	#
.	.	.	.	#
.	#
.	#	#	.	.	.
.	#	.
.
.
.

Each correctly-drawn cell is worth one point. Any background cells left blank get no credit; they need to be marked with '.'.

Discussion:

This question tests algorithms used for basic computer graphics. Because it takes some time to solve and lends well to partial credit, it makes a good problem to be timed for the tiebreaker. The number and spacing of characters matter, so a grid is given for the answer space.

Code Analysis

Exhibition Event

Wisconsin Science Olympiad State Tournament

UW-Stout, April 1, 2016

Team Name: _____

Team Number:

Student Names: _____

DO NOT OPEN THIS TEST until told to do so.

The first question is timed, which may be used to break tie scores. When you are finished with the first question, turn it in immediately to the event supervisor.

Time for question #1: _____

Written test total score: _____ / 100

Tie Breakers: (1) Highest score on question #1, (2) Least time on question #1, (3) Test questions 9, 7, then 11.

1. This first question is timed, which may be used to break tie scores. When you are finished with this question, turn this sheet in immediately to the event supervisor.

What is the printed output of the following program?

```
public void main(){int p=9;q=1,t=0;while(q<p){t+=q;int n=1;  
while(n<=q)System.out.print('X');System.out.print('\n');q+=  
2;}System.out.printf("%d\n",t);}
```

Write your answer below.

/ 20 points

Solution:

X

XXX

XXXX

XXXXXX

16

[extra lines at the end should be left blank]

4 points for each correct "X" line, plus 4 points for the correct final total.

Discussion:

This code is "obfuscated". It is intentionally made compact and difficult to read.

2. What is the printed output of the following program?

```
// Example1.java

public class Example1
{
    public void one ()
    {
        System.out.println("three");
    }

    public void two ()
    {
        System.out.println("two");
    }

    public void three ()
    {
        one();
    }

    public static void main (String[] args)
    {
        three();
        two();
        one();
    }
}
```

Write your answer below.

/ 3 points

Solution:

three

two

three

Incorrect solutions:

threetwothree

The printing is not on separate lines. 2 points.

one

two

three

The first line is incorrect. 2 points for the last two correct lines.

two

three

three

Even though all three lines are present, they are in the wrong order. Only the last line counts, 1 point.

Discussion:

This question tests basic knowledge of method calls and printing. It is a simple enough question that all teams should be able to answer (a good exam should have one such question). It also demonstrates one possible way to score the problem. Because the number and order of output lines matter, horizontal writing lines are provided.

3. What is the printed output of the following program?

```
private int a = 5;  
private int b = 10;  
  
public static void main (String[] args)  
{  
    int result = a + b * 4;  
    System.out.println(result + 3);  
}
```

Write your answer below.

/ 3 points

Solution:

48

Incorrect solutions:

63

This would be the result if expressions were evaluated left-to-right; however, multiplication takes precedence over addition. 1 point.

453

The output should be mathematically added, not concatenated. 1 point.

Discussion:

This question tests integer variables, integer operators, and precedence.

4. What is the printed output of the following program?

```
public static void main (String[] args)
{
    System.out.println(4%8, 14%9, 20%7);
}
```

Write your answer below.

/ 6 points

Solution:

456

(3x2 points per digit. The last 2 lines should be blank, else minus 2.)

Incorrect solutions:

4

5

6

The printing should not be on separate lines. 4 points.

Discussion:

This question tests the modulus operator and the concatenation operation of `System.out.println`. It also demonstrates that it is not necessary to use every line of the answer space.

5. What is the printed output of the following code fragment?

```
String s = "This is a test.";
int i = 0;
while (i < 10) {
    char c = s[i];
    if (c >= 'a' && c <= 'z')
        c = c - 'a' + 'A';
    else if (c >= 'A' && c <= 'Z')
        c = c - 'A' + 'a';
    System.out.print(c);
    i = i + 1;
}
```

Write your answer below.

/ 8 points

Solution:

tHIS 1S A

(7x1 point per non-space character, +1 for correct number of characters.)

Incorrect solutions:

tHIS 1S A TEST.

The loop executes for only 10 characters, not the full length of the string. 7 points.

tHIS iS A

The '\$' and '1' characters are outside the ranges that are tested, and therefore remain unchanged. 6 points.

Discussion:

This question tests knowledge of loops, strings, and character arithmetic. The loop is kept simple for beginning programmers (for example, using `i = i+1` rather than `i++`). Possible variants could include dropping the else, and replacing `>=` with `>`.

6. What is the printed output of the following code fragment?

```
int x = 5;
boolean done = false;
while (!done) {
    System.out.print(x);
    done = (--x < 1);
}
System.out.println(" liftoff!");
```

Write your answer below.

/ 8 points

Solution:

54321 liftoff!

(5x1 point per digit, +2 points for correct number of digits, +1 for liftoff!)

Incorrect solutions:

543210 liftoff!

The loop should terminate as soon as x becomes 0. 6 points.

Discussion:

This question tests boolean variables, the logical not operator, while loops, and predecrement.

Possible variants could include starting with done = true, or using a postdecrement.

7. What is the printed output of the following code fragment?

```
int[] fibs = { 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 };  
int q = 10;  
int lo = 0; hi = 8;  
while (lo < hi) {  
    int mid = (lo+hi) / 2;  
    if (q < fibs[mid])  
        hi = mid;  
    else  
        lo = mid;  
}  
System.out.println(lo);
```

Write your answer below.

/ 12 points

Solution:

5

Incorrect solutions:

6

When the searched value is between two array values, the program should print the *lower* index.
8 points.

8

Prints the index, not the array value. 8 points.

13

Both of the above mistakes. 4 points.

Discussion:

This is a binary search. It also tests knowledge of arrays.

8. What is the printed output of the following code fragment?

```
String a = "SCIENCE OLYMPIAD";
String b = a.charAt(a.length - 2);
String c = a.substring(8, 1);
String d = a.substring(3, 2);
String e = d.substring(1, 1).concat(d.substring(0, 1));
System.out.println(b, c, e);
```

Write your answer below.

/ 8 points

Solution:

DONE

2 points for each letter.

Discussion:

This question tests knowledge of string operations.

9. What is the printed output of the code below when it is called as `f(12)`?

```
private int primes[] = { 2, 3, 5, 7, 11, 13, 17, 19 };

public void f (int n)
{
    int i = 0;

    while (i < 8) {
        int d = primes[i];
        if ((n % d) == 0) {
            System.out.printf("%d ", d);
            f(n/d);
            return;
        }
        i = i + 1;
    }
    System.out.print("\n");
    return;
}
```

Write your answer below.

/ 12 points

Solution:

2 2 3
(4x3 points for each term)

Incorrect solutions:

2 3

Each time the method is called, it restarts the factor search. 8 points.

Discussion:

This is an example of a recursive method. It also shows how a problem can be specified as a method call – in this case, f(12).

10. What is the printed output of the following program?

```
// Scope.java

public class Scope
{
    private static int x = 1;

    public static void main(String[] args)
    {
        int x = 2;

        System.out.printf("%d ", x);
        a();
        b();
        a();
        b();
        System.out.printf("%d ", x);
    }

    public static void a()
    {
        int x = 3;

        System.out.printf("%d ", x++);
        System.out.printf("%d ", x);
    }

    public static void b()
    {
        System.out.printf("%d ", x);
        x *= 10;
        System.out.printf("%d ", x);
    }
}
```

Write your answer below.

/ 10 points

Solution:

2 3 4 1 10 3 4 10 100 2
1 point per correct answer.

Discussion:

This question tests variable scope. Because a horizontal answer blank is given, the spacing of the answer does not matter.

11. What is the printed output of the following code fragment?

```
private int array[] = { 15, 4, 16, 3, 9, 11, 1, 12, 19, 6 };

int x = 0;
while (x < 9) {
    int y = x+1;
    while (y < 10) {
        if (array[x] > array[y]) {
            // Swap the values of x and y
            int tmp = array[x];
            array[x] = array[y];
            array[y] = tmp;
        }
        y = y + 1;
    }
    x = x + 1;
}

// Print the array
int i = 0;
while (i < 10) {
    System.out.printf("%d ", array[i]);
    i = i + 1;
}
```

Write your answer below.

/ 10 points

Solution:

1 3 4 6 9 11 12 15 16 19

1 point per number that is greater than the previous number.

Discussion:

This is an example of sorting (a bubble sort).

संदर्भ संख्या : 92214000006926 , दिनांक - 27 Apr 2022 तक की स्थिति

आवेदनकर्ता का विवरण :

शिकायत संख्या:-

92214000006926

आवेदक का नाम-

राम गोयल

विषय-

आवेदक द्वारा बताया गया कि छात्र का नाम राम गोयल , पिता का नाम अतुल कुमार गोयल , जाति वर्ग सामान्य , रजिस्ट्रेशन संख्या 090550502100523 , जन्म तिथि 24/8/2002 , कॉलेज का नाम ABES इंजीनियरिंग कॉलेज , कक्षा कम्प्यूटर साइंस एंड इंजीनियरिंग द्वितीय वर्ष , वर्ष 2021,2022 , कॉलेज का जिला गाजियाबाद ,आवेदक को छात्रवृत्ति नहीं मिली है , कृप्या जल्द से जल्द समाधान किया जाये

विभाग -

समाज कल्याण विभाग

शिकायत श्रेणी -

नियोजित तारीख-

07-05-2022

शिकायत की स्थिति-

स्तर -

जनपद स्तर

पद -

जिला समाज कल्याण अधिकारी

प्राप्त रिमाइंडर-

प्राप्त फीडबैक -

दिनांक को फीडबैक:-

फीडबैक की स्थिति -

संलग्नक देखें -

[Click here](#)

नोट- अंतिम कॉलम में वर्णित सन्दर्भ की स्थिति कॉलम-5 में अंकित अधिकारी के स्तर पर हुयी कार्यवाही दर्शाता है!

अग्रसारित विवरण :

क्र.स.	प्राप्त दिनांक	नियत दिनांक	अधिकारी को प्रेषित	आदेश	आख्या दिनांक	आख्या रिपोर्ट	स्थिति	संलग्नक
1	22-04-2022	07-05-2022	जिला समाज कल्याण अधिकारी- गाजियाबाद, समाज कल्याण विभाग				कार्यालय स्तर पर लंबित	

How can the course file method be made smoother?

1. Problem Statement:

A course file is essentially a record that contains all types of relevant facts about the batch, assessment, and overall outcomes of the course in an academic setup. Course File is basically the “**Geeta**” of the faculty member for a particular batch he/she is teaching.

A course file provides you with a leg up on the competition when it comes to the course's overall curriculum and administration. A course file will provide you with all of the information you require to make an informed selection like COs, CO and PO mapping, Target of COs and attainment, Lecture Plan, Lecture delivery schedule, Assignments, Sessional Papers and End Term papers. The data you have about the course and students are used to determine delivery mechanisms, change or expand the curriculum, provide more learning materials, use different teaching styles, and so on.

Faculty members are normally required to retain a course file at all universities/colleges. **Maintaining a hard copy of one, however, is a very difficult task as there are a lot of documents to include, a lot of data to analyze, and a lot of time to put in.**

2. Existing Course Files Contents:

S.no.	CONTENT
PART A	
1.	ABES Mission, Vision statement & Quality Policy – <i>one copy</i>
2.	Department Mission & Vision – <i>one copy</i>
3.	List of Students – (<i>double column format</i>)
4.	PEOs, POs, PSOs, Course Outcomes and Mapping with POs/PSOs – <i>one copy</i>
5.	Last year CO attainment and suggested action- <i>one copy</i>
6.	Academic calendar – <i>one copy</i>
7.	University Evaluation Scheme – <i>one copy</i>
8.	University Subject Syllabus – <i>one copy</i>
9.	Class Time Table – <i>one copy</i>
10.	Faculty Time Table – <i>one copy</i>
11.	Lecture Plan along with Sessional Test Schedule (Full Unit/ Half Unit) – <i>one copy</i>
12.	Lecture Delivery Schedule with plan summary of assignments/Test/Quiz
13.	Assignments/ Test/ Quiz with Answer key – (<i>with CO & KL mapping</i>)
14.	Assignment/ Test/ Quiz Marks & CO Attainment
15.	Tutorial sheets for numerical problems
16.	Sessional Tests/ Pre-University Exam Question Papers – (<i>With solution for numerical problems</i>)
17.	Sessional Tests/ Pre-University Exam - Award Sheet
18.	Sessional Tests/ Pre-University Exam - Gap Analysis and CO attainment

19.	List of Weak Students and their Make-up Classes (<i>after each sessional exam</i>)
20.	Review of University Question Paper (<i>and proofs of correspondence with University in case of any discrepancy</i>)
21.	Attendance Sheets – Overall
22.	Evaluated Answer Scripts – (<i>2 Samples</i>)
23.	Course Exit Survey and analysis
24.	CO attainment and recommendations

PART B

1.	Faculty Notes (Handout, PPTs in Separate File) – (<i>for each faculty; common ppt/ notes to be marked accordingly</i>)
2.	Books ,web content, other than Text Book frequently referred to – (<i>faculty wise</i>)
3.	Previous Years Question Papers (UPTU - 4 Years) → Descending Order – <i>one copy each</i>

Note: Contents in Part A are to be placed section wise

Note: Proposal is being prepared considering all the basic minimum points available in the course file format of ABESEC.

3. Team formation:

Prof. Divya Mishra (HOD-CSE)

Mr. Ravi Kumar

Mr. Prabhat Singh

Mr. Sunil Kumar

Ms. Jasmine

4. Proposed Solution:

Course File Automation System will take care of all the diverse needs and requirements of course file creation. So regardless of the university or institution, it will be able to produce the course file the way we want.

Faculty members will be able to create a course file with all of the necessary information with a single click. **Faculty members can print out a custom designed course file in a few minutes rather than spending hours gathering and processing data.** Simply select which data you want to include in the course file and click the button.

As we all know, course files are generated for each Theory and Lab subject for each batch.

Step by Step Process to Print a Course file:

Step 1: As we log in to our faculty account, we will have a comprehensive view of all the batches we teach as well as the batches that belong to our department.

Step 2: Choose a batch here to develop a course file for the subject we teach for that batch.

Step 3: Once we enter the batch, the course file settings can be found on the menu. Simply click that, we can pick the details we want in our course file (samples/formats of LDS, CO PO Mappings etc. Will be added), and then click the course file option down below.

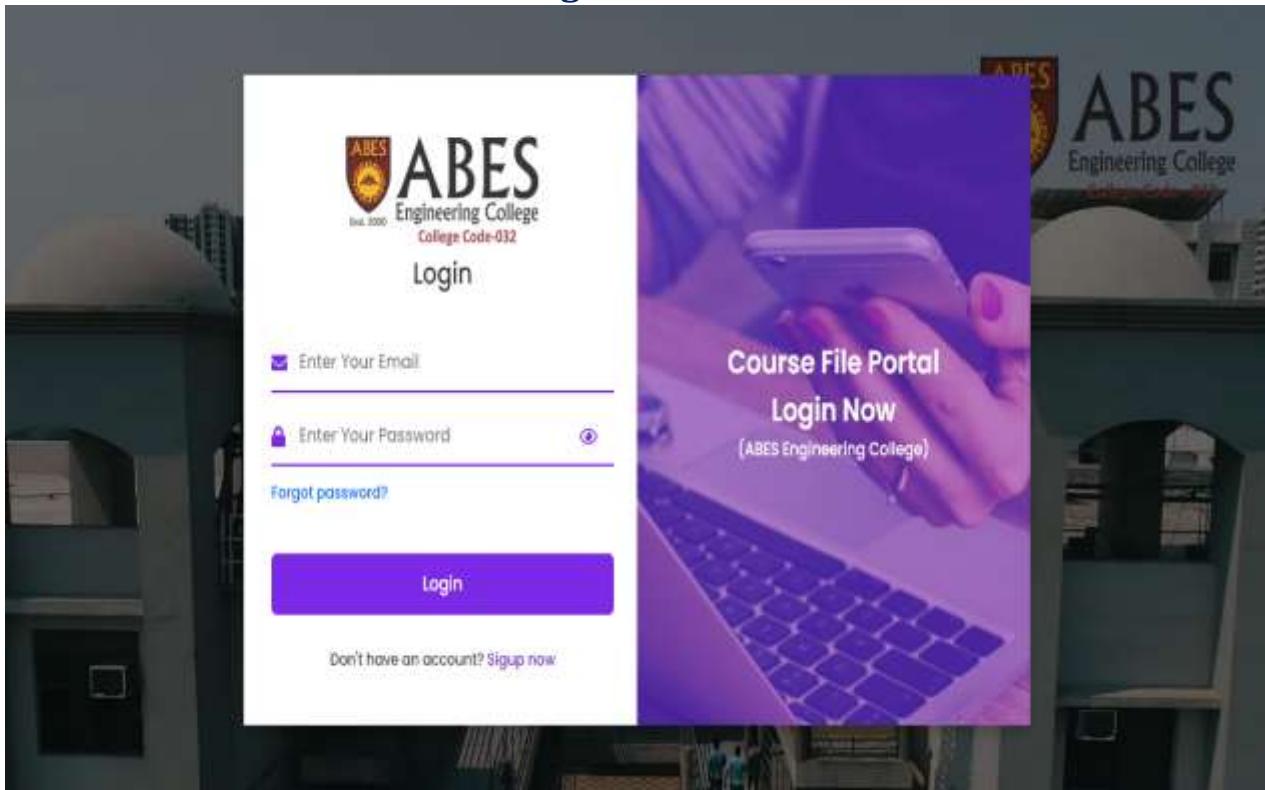
Step 4: To make the procedure easy, we can add details like Vision, Mission, Program Objectives and also options to add course objective, course outcomes, syllabus, and all will be given. To include such details, we won't have to go anywhere else.

Step 5: After creation, just click on the print button to get the course file.

That's how simple it'll become to get a course file.

5. Course File Portal Working Module (Step by Step Screenshot):

Login Screen



Content 1- ABES Mission and Vision Statement

A screenshot of the ABES Course File Portal showing the 'Create a File' section. The interface includes a sidebar with icons for Home, Dashboard, View File, and Logout. The main area has a blue header bar with a user profile icon and the name 'Ram (12345)'. The 'Create a File' form has several input fields: 'DAA Course File' (with 'RCS-502' and '2020-21' listed), 'Event' (with 'Even'), and an 'Upload deptMissVision File' section where a PDF file named 'Vision Mission College-converted.pdf' has been uploaded. A 'Create Course File' button is at the bottom of the form.

Content 2- Department Mission and Vision Statement

The screenshot shows a 'Create a File' form. The left sidebar includes icons for Home, Dashboard, View Files, and Logout. The main form fields are: DAA Course File (RCS-502), RCS-502, 2020-21, Even, Upload deptMissVision File (Browse... Dep Mission Vision.pdf, Create Course File button, message: * Dep Mission Vision.pdf Uploaded).

Content 3- List of Students

The screenshot shows a 'Create a File' form. The left sidebar includes icons for Home, Dashboard, View Files, and Logout. The main form fields are: DAA Course File (RCS-502), RCS-502, 2020-21, Even, Upload studentList File (Browse... Student List.pdf, Create Course File button, message: * Student List.pdf Uploaded).

Content 4 - PEO's, POs, PSOs of the Department

The screenshot shows a 'Create a File' form. The left sidebar includes icons for Home, Dashboard, View Files, and Logout. The main form fields are: DAA Course File (RCS-502), RCS-502, 2020-21, Even, Upload acadCalender File (Browse... 2. Vission Mission PEO PO PSOs.pdf, Create Course File button, message: * 2. Vission Mission PEO PO PSOs.pdf Uploaded).

Content 5 - Prerequisite, Course Outcome and Mapping of CO with POs/PSOs

The screenshot shows a user interface for creating a course file. On the left is a vertical sidebar with icons for Home, Dashboard, View File, and Logout. The main area has a blue header with a logo and the name "Ram (12345)". Below the header, the title "Create a File" is displayed. The form fields include:

- DAA Course File: RCS-502
- Year: 2020-21
- Semester: Even
- Upload CO-PO Mapping File:
 - Browse...: CO PO Mapping Format DAA-1.pdf
 - Create Course File
- A message: * CO PO Mapping Format DAA-1.pdf Uploaded.

Content 7- Academic Calendar

The screenshot shows a user interface for creating an academic calendar. The layout is identical to the Content 5 page, with a sidebar on the left and a main form area. The form fields include:

- DAA Course File: RCS-502
- Year: 2020-21
- Semester: Even
- Upload acadCalender File:
 - Browse...: ACADEMIC CALENDAR(Odd Sem. 2018-19).pdf
 - Create Course File
- A message: * ACADEMIC CALENDAR(Odd Sem. 2018-19).pdf Uploaded.

Content 8- University Syllabus

This screenshot shows the 'Create a File' interface for a university syllabus. The left sidebar includes icons for Home, Dashboard, View Files, and Logout. The main area has fields for 'DAA Course File' (set to 'RCS-502'), 'Year' (set to '2020-21'), and 'Semester' (set to 'Even'). Below these is a file upload section for 'uniSyllabus File'. A 'Browse...' button shows 'DAA-Syllabus.pdf' selected. A 'Create Course File' button is present. A success message at the bottom states '+ DAA-Syllabus.pdf uploaded'.

Content 11- Lecture Plan

This screenshot shows the 'Create a File' interface for a lecture plan. The left sidebar includes icons for Home, Dashboard, View Files, and Logout. The main area has fields for 'DAA Course File' (set to 'RCS-502'), 'Year' (set to '2020-21'), and 'Semester' (set to 'Even'). Below these is a file upload section for 'lecPlan File'. A 'Browse...' button shows 'DAA LP.pdf' selected. A 'Create Course File' button is present. A success message at the bottom states '+ DAA LP.pdf Uploaded'.

Content 12 - Lecture Delivery Schedule

The screenshot shows a user interface for creating a course file. On the left is a vertical sidebar with icons for Home, Dashboard, View Files, and Logout. The main area has a blue header bar with a logo and the name "Ram (12345)". Below the header, the title "Create a File" is displayed. There are four input fields: "DAA Course File" containing "RCS-502", "2020-21", and "Even". Below these is a section titled "Upload LecDeliverySchedule File" with a "Browse..." button showing "DAA LDS.pdf" and a "Create Course File" button. A message at the bottom indicates the file was uploaded successfully.

Content 13- Assignment/Test/Quiz with Answer Key

This screenshot shows the same "Create a File" interface as the previous one. The sidebar and header are identical. The main area has the same four input fields: "DAA Course File" (RCS-502, 2020-21, Even). The "Upload Assignments File" section shows a "Browse..." button with "Assignment 1 DAA Odd Sem 2019-20.pdf" and a "Create Course File" button. A message at the bottom confirms the file was uploaded.

Content 14 - Assignment/Test/Quiz Marks

The screenshot shows a user interface for creating a file. At the top, there is a blue header bar with the title "Content 14 - Assignment/Test/Quiz Marks". On the right side of the header is a user profile icon labeled "Ram (12345)". Below the header is a sidebar on the left containing icons for "Home", "Dashboard", "View Files", and "Logout". The main content area has a title "Create a File" and several input fields:

- "DAA Course File": A dropdown menu showing "RCS-502", "2020-21", and "Even".
- "Upload assignmentMarks File": A file upload section with a "Browse..." button, a file path "Assignment Marks.pdf", and a "Create Course File" button.
- A message box at the bottom indicates that "Assignment Marks.pdf Uploaded".

Content 23 - Course Exit Survey

The screenshot shows a user interface for creating a file. At the top, there is a blue header bar with the title "Content 23 - Course Exit Survey". On the right side of the header is a user profile icon labeled "Ram (12345)". Below the header is a sidebar on the left containing icons for "Home", "Dashboard", "View Files", and "Logout". The main content area has a title "Create a File" and several input fields:

- "DAA Course File": A dropdown menu showing "RCS-502", "2020-21", and "Even".
- "Upload courseExitSurvey File": A file upload section with a "Browse..." button, a file path "Course Exit Survey Section A.pdf", and a "Create Course File" button.
- A message box at the bottom indicates that "Course Exit Survey Section A.pdf Uploaded".

Uploading Files Together:

The screenshot shows a user interface for creating a file. At the top right is a profile icon for 'Ram (12345)'. Below it, the title 'Create a File' is displayed. There are four input fields containing 'DAA Course File', 'RCS-502', '2020-21', and 'Even'. A section titled 'Upload sessQuestionPaper File' contains a 'Browse...' button followed by the message '13 files selected.' and a 'Create Course File' button. A list of 13 uploaded files is shown below, each with a small preview icon.

- 2. Vission Mission PEO PO PSOs.pdf Uploaded
- ACADEMIC CALENDAR(Odd Sem. 2018-19).pdf Uploaded
- Assignment 1 DAA Odd Sem 2019-20.pdf Uploaded
- Assignment Marks.pdf Uploaded
- CO PO Mapping Format DAA-1.pdf Uploaded
- Course Exit Survey Section A.pdf Uploaded
- DAA LDS.pdf Uploaded
- DAA LP.pdf Uploaded
- DAA Syllabus.pdf Uploaded
- Dep Mission Vission.pdf Uploaded
- PEOPSO.pdf Uploaded
- Student List.pdf Uploaded
- Vission Mission College-converted.pdf Uploaded

Viewing Uploaded Files:

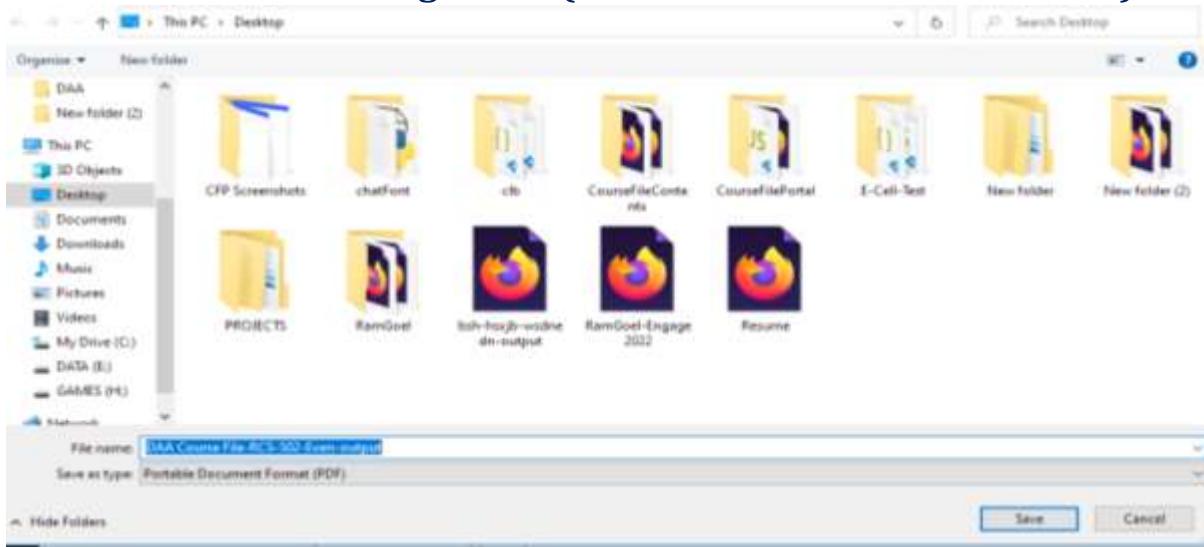
The screenshot shows a list of uploaded files under the heading 'All Files'. On the left is a vertical sidebar with icons for Home, Dashboard, View File, and Logout. The main area lists five files with their details and three action icons (trash, download, and share) to the right of each entry.

File Name	Uploader	Action
2. Vission Mission PEO PO PSOs.pdf	Ram	
ACADEMIC CALENDAR(Odd Sem. 2018-19).pdf	Ram	
Assignment 1 DAA Odd Sem 2019-20.pdf	Ram	
Assignment Marks.pdf	Ram	
CO PO Mapping Format DAA-1.pdf	Ram	

If No Files are Uploaded



Download of Merged File (of Name Provided in Dashboard)



Merged Course File Preview

The screenshot displays a merged course file preview across three panels. Each panel contains the ABES Engineering College logo and the title "ABES Engineering College, Ghasialbad Department of Computer Science & Engineering".

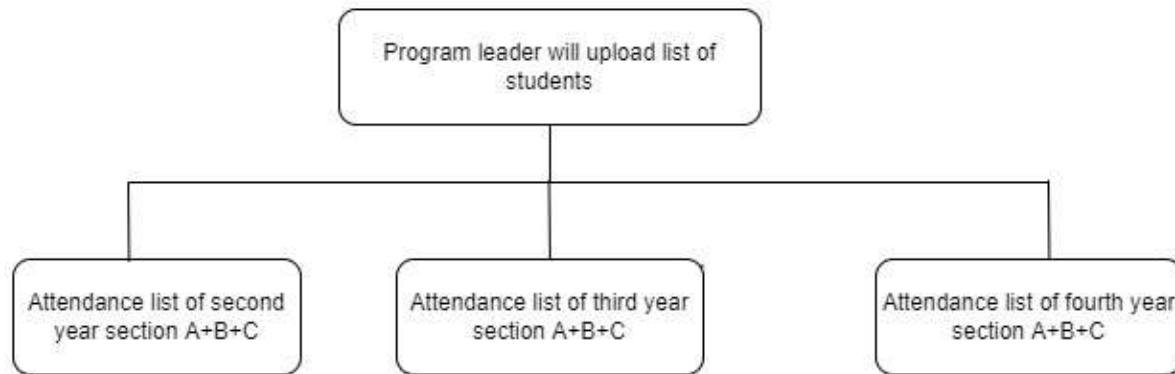
- Panel 1:** Contains sections for "Vision and Mission of the Department" and "Aims of Course".
- Panel 2:** Contains sections for "Programme Educational Objectives (PEOs)" and "Programme Outcomes".
- Panel 3:** Contains sections for "Programme Outcomes" and "Programme Learning Outcomes (PLOs)".

The content is identical across all panels, reflecting the merged nature of the document.

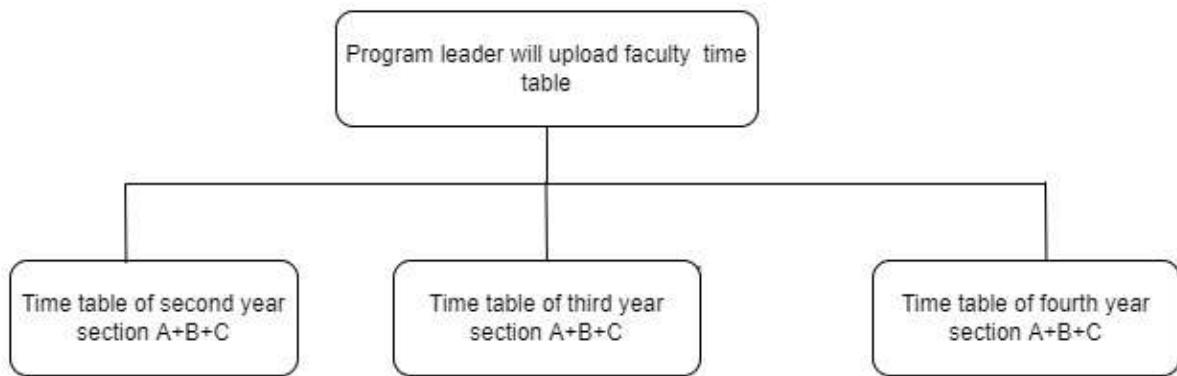
Full PDF File - [Course File Document](#)

6. Work Flow Diagram for maintaining common minimum contents of Course Files at a single place (Sample): Access available to all the faculty members of the department in order to avoid duplicate work.

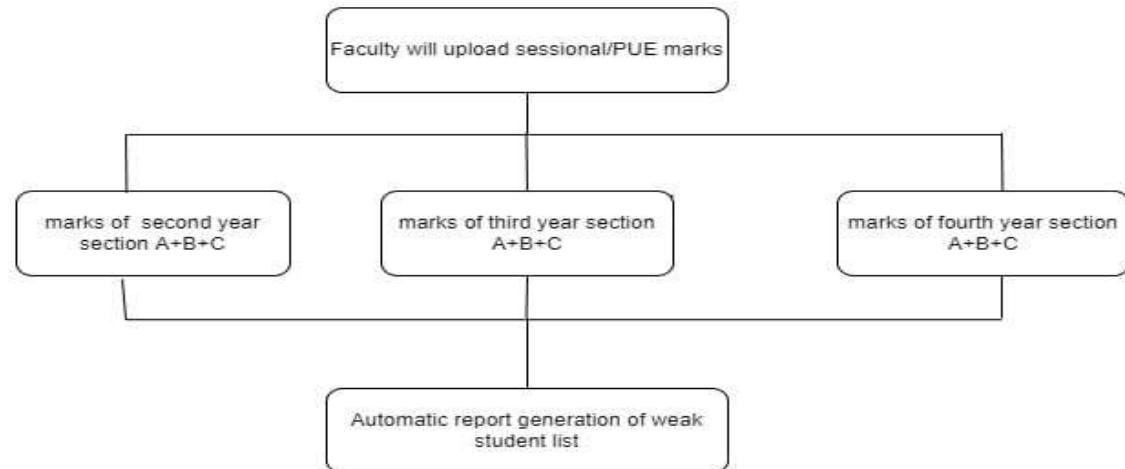
a. Uploading of List of Students on portal:



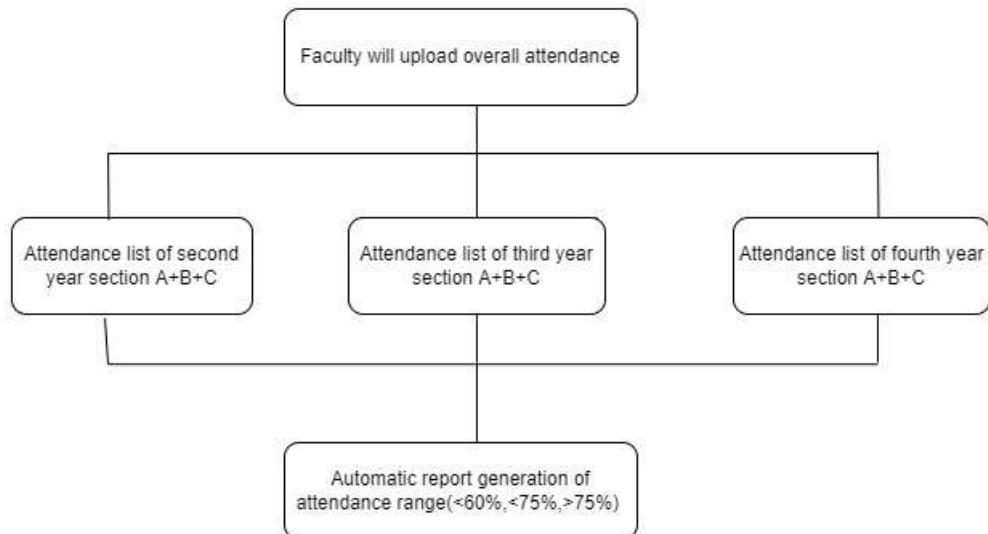
b. Uploading of Time table on portal:



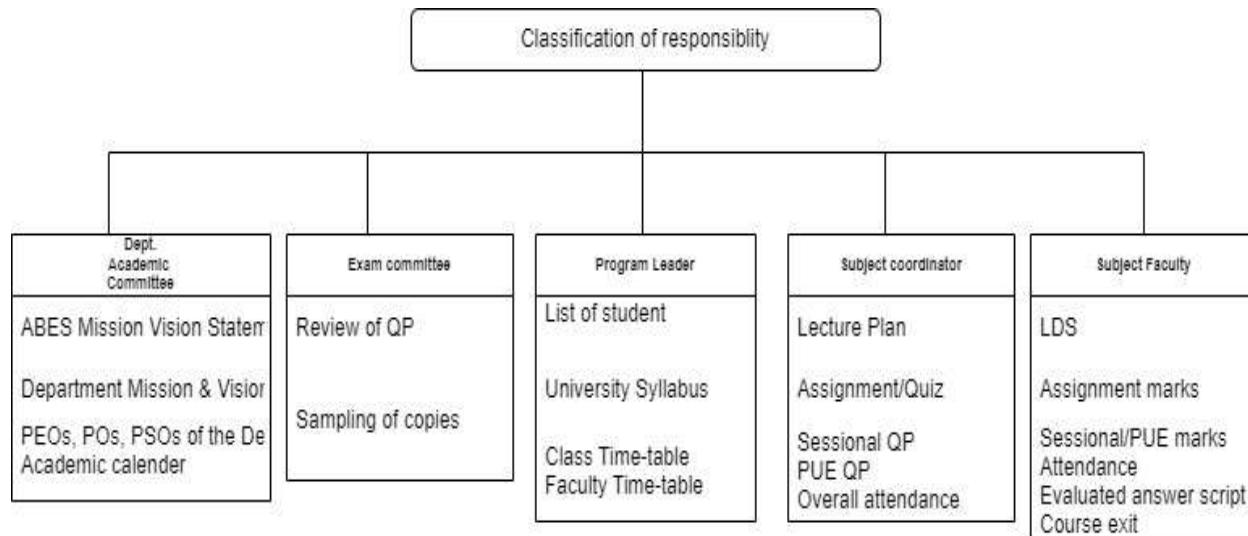
c. Uploading of Sessional Marks on portal:



d. Uploading of Attendance on portal:



e. Classification of responsibility seeking approval for the contents of course files:



7. Smooth ride for Faculty members from Course File Automation Portal: Outcome

1. **Less paper Consumption:** Extra consumption of paper will be controlled.
2. **Data Loss Avoidance:** Faculty data will always be saved on the portal for accessibility
3. **Time saving:** No need to create course file from scratch always. Already created course file content can be reused.
4. **No need for data formatting.** Course file will be downloaded in all the formats like pdf, .xls etc.
5. **In-house project of ABESEC:** Students are working on the project. Therefore, it will be beneficial for students as well as for faculties.
6. **Scalability:** More features can be added as per demand.
7. **Interactive and User friendly.**
8. **Anywhere and anytime accessibility.**

8. Demonstration Video of Course File Automation portal: [Demonstration Video Course File Automation](#)

कार्यालय जिला समाज कल्याण अधिकारी, गाजियाबाद।
पत्रांक 733 /आई0जी0आर0एस0/2021-22 २७ दिनांक 104/2022
शिकायत सन्दर्भ संख्या— 92214000006926

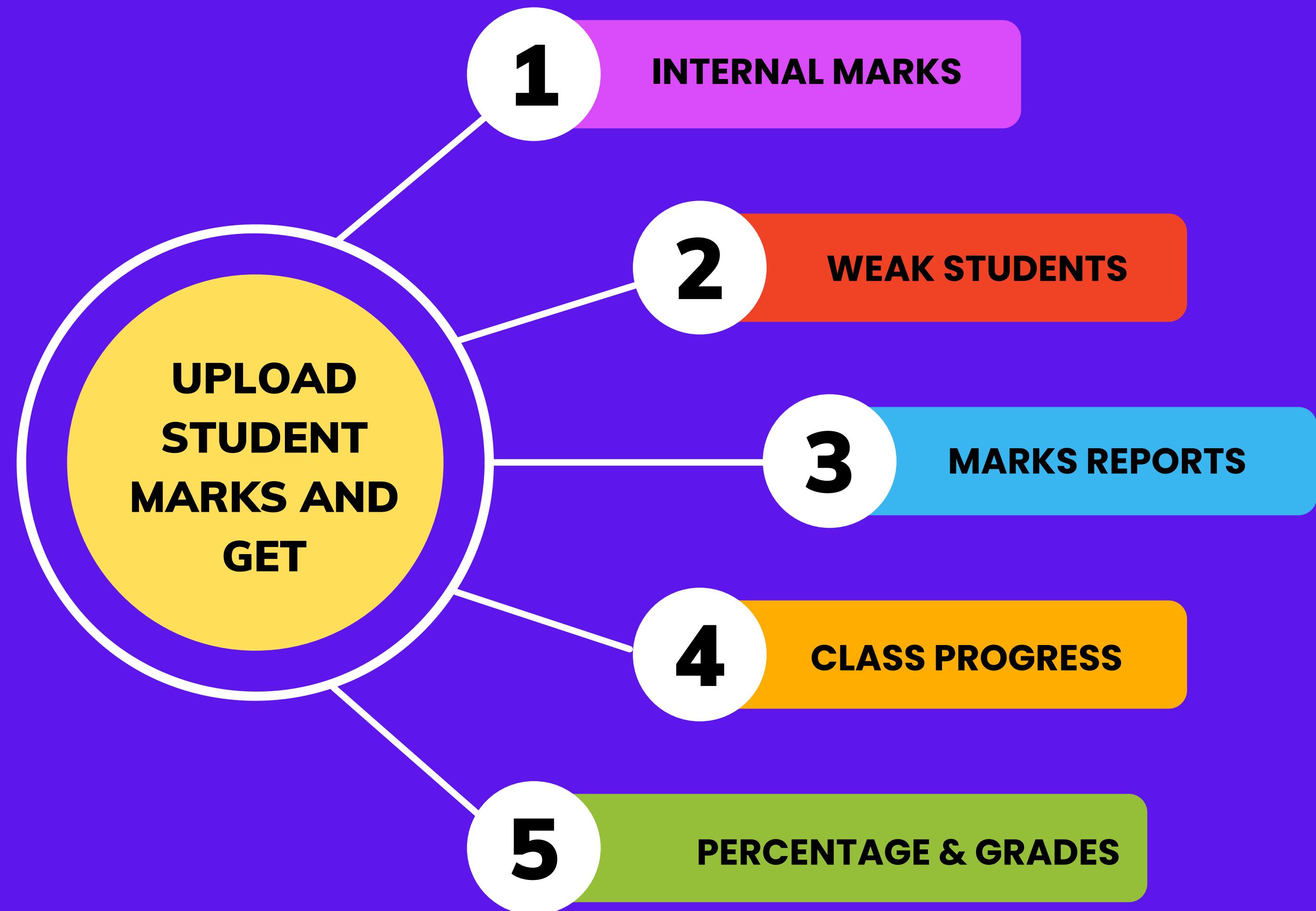
जांच अधिकारी का नाम	संजीव कुमार
जांच अधिकारी का पदनाम	प्रधान सहायक
फोन नं०	01202985875
शिकायत कर्ता का नाम व पता	राम गोयल
शिकायतकर्ता का मोबाइल नं०	—
शिकायत का स्थल	गाजियाबाद
सुनवाई निस्तारण का दिनांक	25 अप्रैल, 2022
अधिकारी की आख्या	छात्रवृत्ति बेबसाइट पर वर्ष 2021-22 की छात्रवृत्ति/ शुल्क प्रतिपूर्ति के अन्तर्गत छात्र का आवेदन पत्र जनपद स्तर पर से बैरीफाई किया गया है। मुख्यालय लखनऊ स्तर से योजनात्तर्गत पर्याप्त धनावंटन उपलब्ध न होने के कारण नियमानुसार अस्वीकृत कर दिया गया है, जिसके कारण छात्रवृत्ति धनराशि देय नहीं है।
	कृ० प्रकरण निष्केपित करने का कष्ट करें।

जिला समाज कल्याण अधिकारी
गाजियाबाद

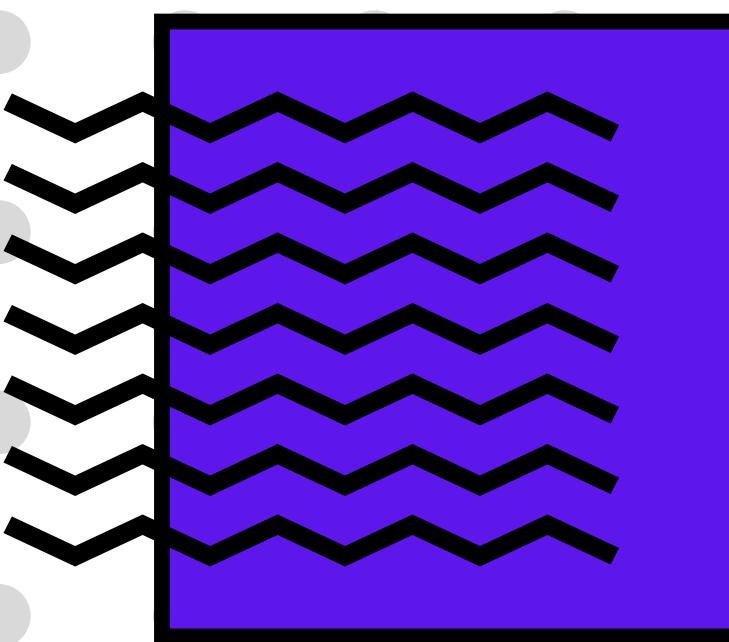
प्रतिलिपि—

- 1—जिलाधिकारी महोदय, गाजियाबाद को सादर सूचनार्थ प्रेषित।
- 2—सम्बन्धित शिकायतकर्ता ।

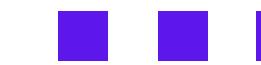
जिला समाज कल्याण अधिकारी
गाजियाबाद



ABES ENGINEERING COLLEGE



COURSE FILE PORTAL

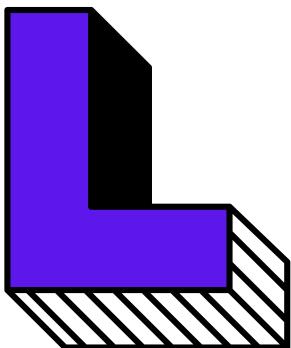


Entrepreneurship Cell, CSE Department, ABESEC



WHAT IS A COURSE FILE ?

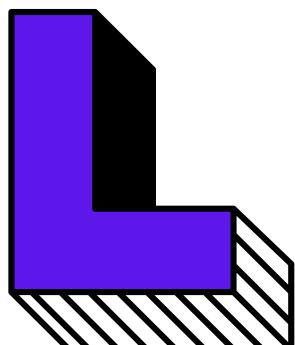
A course file is essentially a record that contains all types of relevant facts about the batch, assessment, and overall outcomes of the course in an academic setup. A course file provides you with a leg up on the competition when it comes to the course's overall curriculum and administration. A course file will provide you with all of the information you require to make an informed selection like COs, CO and PO mapping, Target of COs and attainment, Lecture Plan, Lecture delivery schedule, Assignments, Sessional Papers and End Term papers. The data you have about the course and students are used to determine delivery mechanisms, change or expand the curriculum, provide more learning materials, use different teaching styles, and so on. Faculty members are normally required to retain a course file at all universities/colleges.





PROBLEM STATEMENT

Maintaining a hard copy of one, however, is a very difficult task as there are a lot of documents to include, a lot of data to analyze, and a lot of time to put in. and also a lot of paper gets waste in making and maintenance of physical course file, and the content of one course file can not be reused for others.





WHY COURSE FILE PORTAL

01

PAPER WASTAGE

When the course file goes online, it will reduce the paper wastage .

02

UNSTRUCTURED DATA

All data will be available at one place and in an structured way.

03

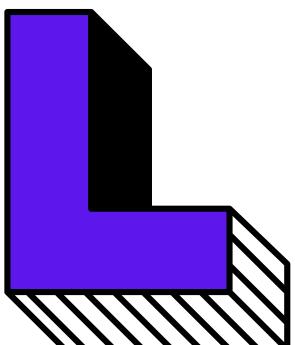
MANUAL TASKS

The manual tasks done by the faculties will be done automatically.

04

MAINTAINANCE

It will become easy to manage, access and share all course files.





Hi, I am a faculty and
I need a course file
for every semester.

I prepare it
manually and
it takes lot of
pages.



Is there any
solution for
this.



Sir We can solve
your problem by
making a online
portal.



Oh, can you tell
me how a portal
can solve my
problem.



This portal will
automate all the work
of course file and will
make it easy.

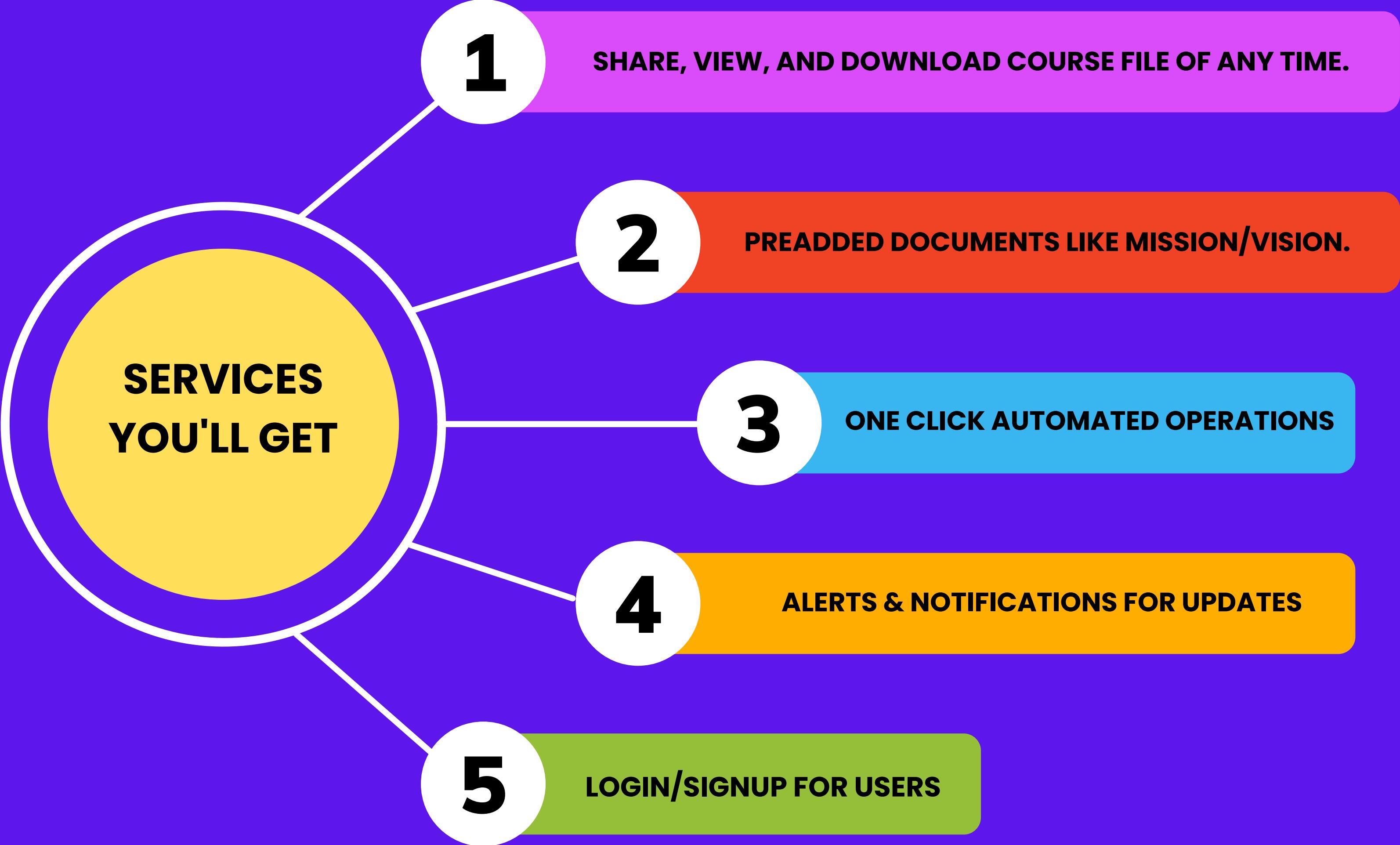
Let us explain more



What/How of Portal?

The portal is smart.





SERVICES YOU'LL GET

1

SHARE, VIEW, AND DOWNLOAD COURSE FILE OF ANY TIME.

2

PREADDED DOCUMENTS LIKE MISSION/VISION.

3

ONE CLICK AUTOMATED OPERATIONS

4

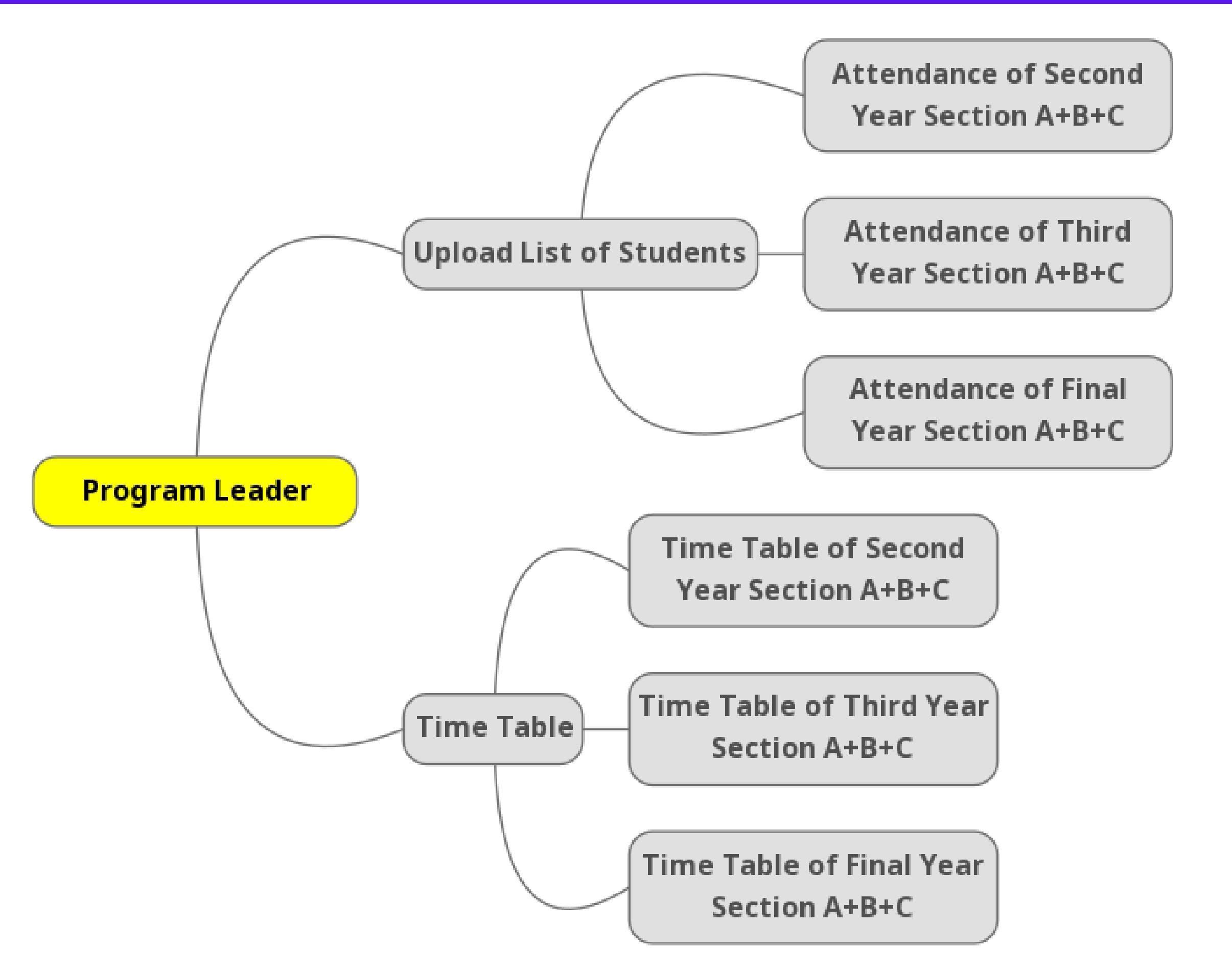
ALERTS & NOTIFICATIONS FOR UPDATES

5

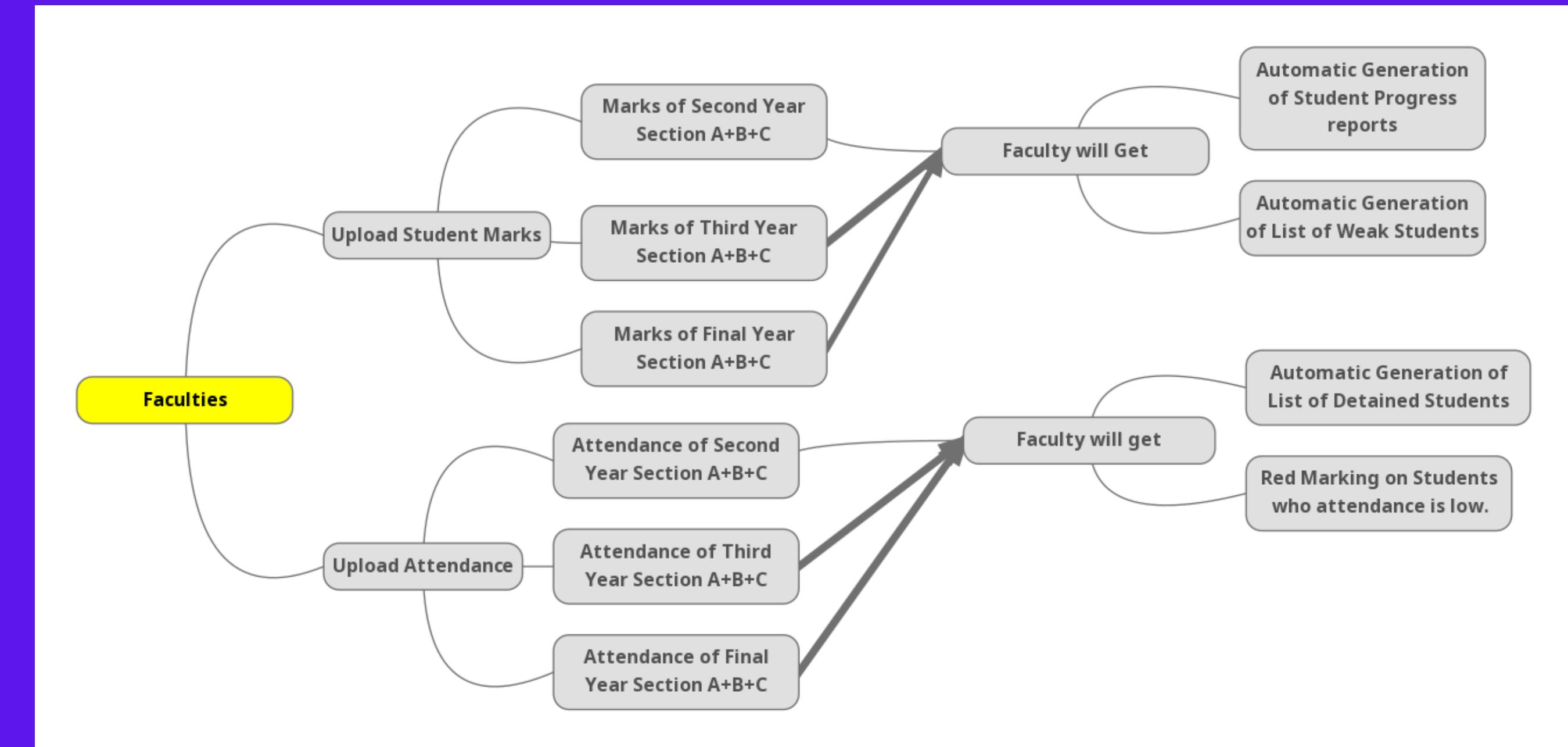
LOGIN/SIGNUP FOR USERS

DATA FLOW IN PORTAL

(Make Templates)



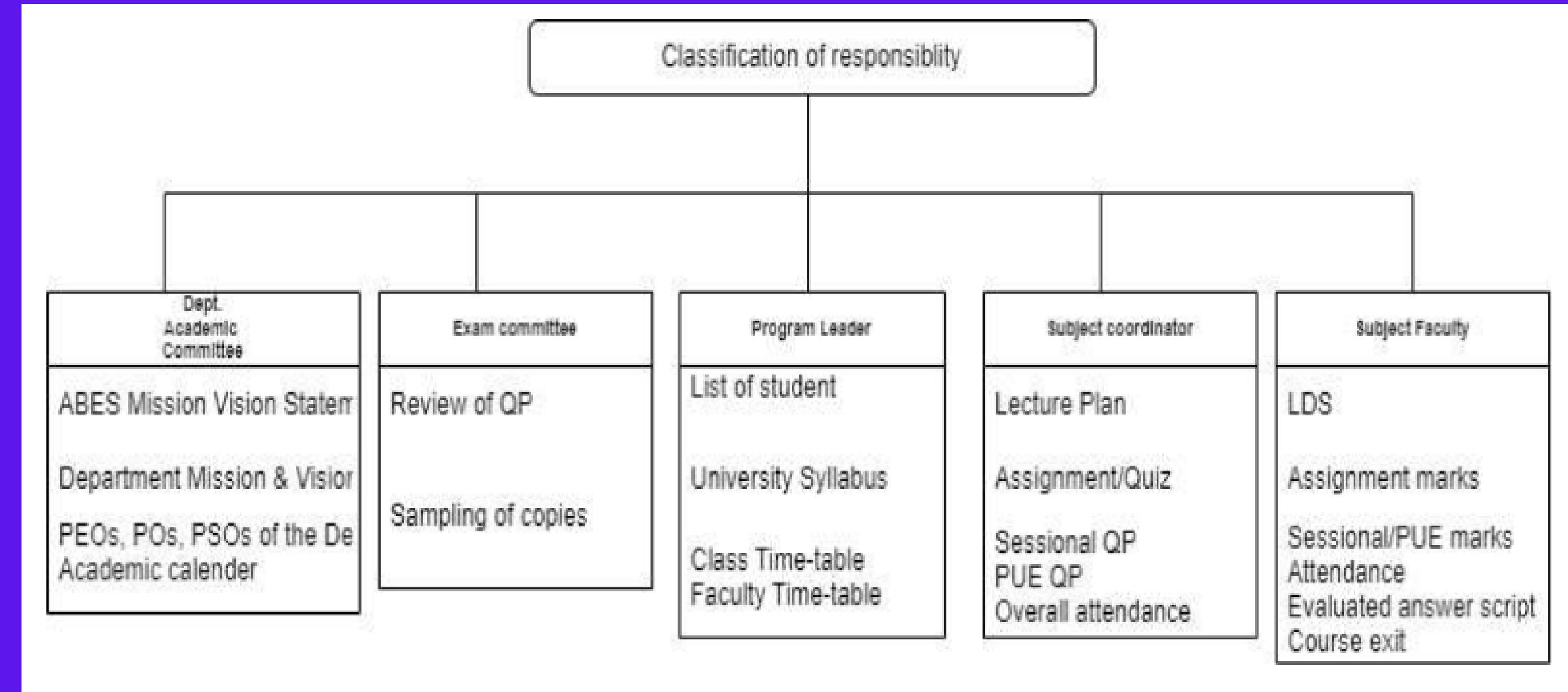
DATAFLOW IN PORTAL (Add Data)



DATAFLOW

IN PORTAL

(Uploading of Data)



WorkFlow

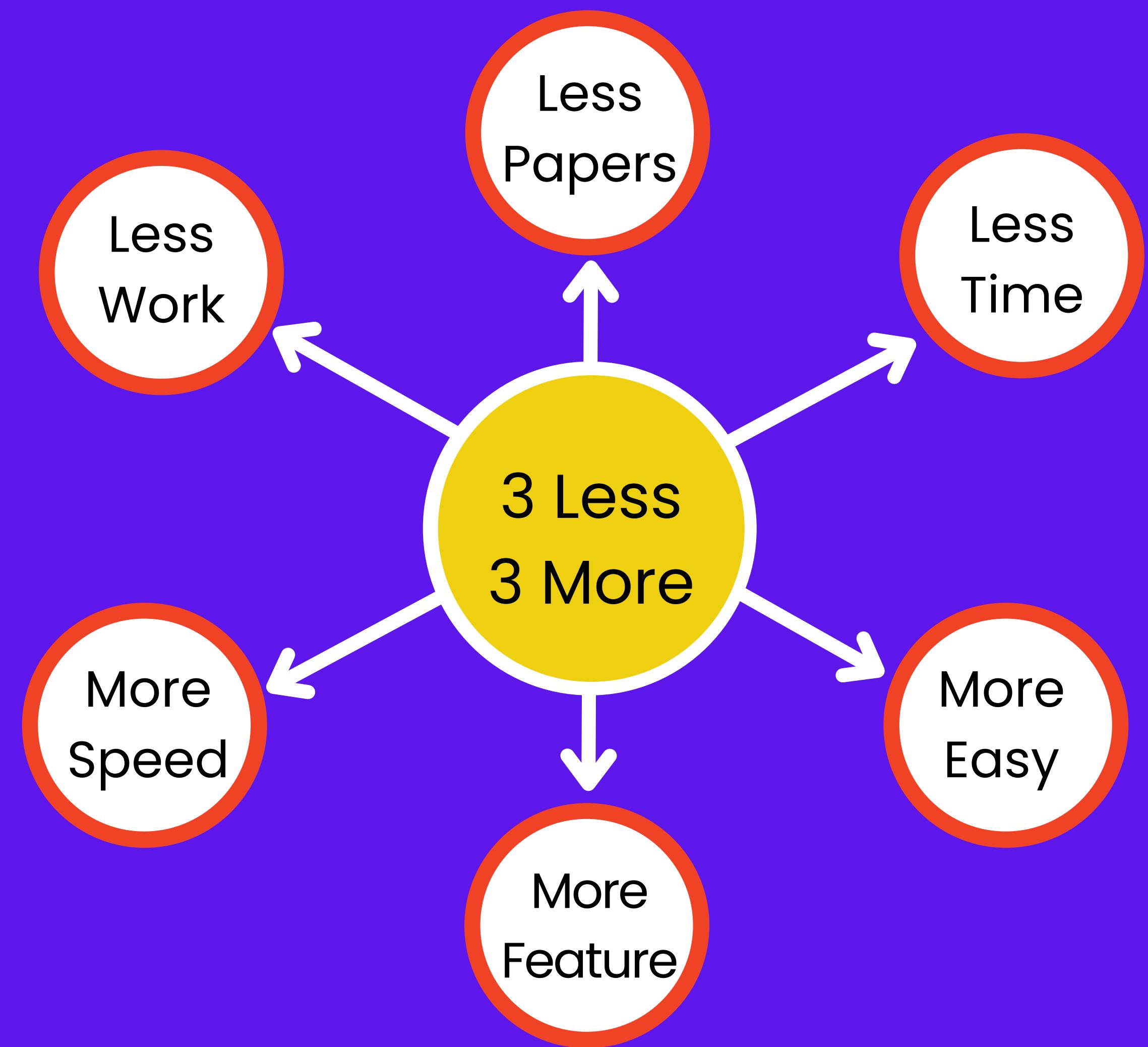
Login into Account

Choose the Batch

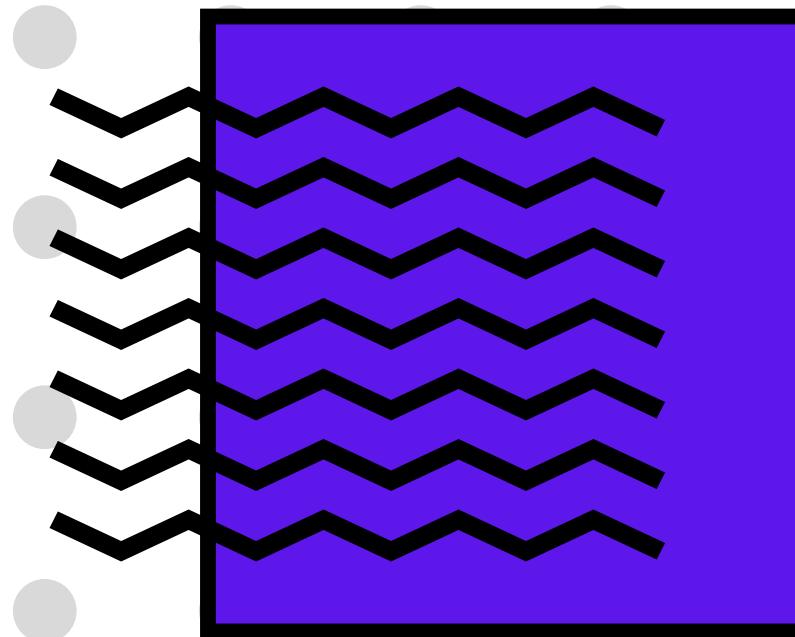
Select Default Files

Add the Required Data

The File is Created



ABES ENGINEERING COLLEGE



**THANKS FOR
BEING HERE**



**UPLOAD
STUDENT
ATTENDANCE
AND GET**

The diagram illustrates a three-step process for managing student attendance. It begins with a large yellow circle containing the text "UPLOAD STUDENT ATTENDANCE AND GET". Three white lines extend from the bottom of this circle to three numbered steps: "1 DETAINED LIST" (purple bar), "2 HIGHLIGHT ON LESS ATTENDANCE" (orange bar), and "3 ATTENDANCE REPORTS" (blue bar). Each step is enclosed in a white circle with a black number.

DETAINED LIST

HIGHLIGHT ON LESS ATTENDANCE

ATTENDANCE REPORTS

1

DETAINED LIST

2

HIGHLIGHT ON LESS ATTENDANCE

3

ATTENDANCE REPORTS

AIM: To write a C program to implement the FCFS scheduling algorithm.

Theory : Given n processes with their burst times, the task is to find average waiting time and average turnaround time using FCFS scheduling algorithm.

First in, first out (FIFO), also known as first come, first served (FCFS), is the simplest scheduling algorithm. FIFO simply queues processes in the order that they arrive in the ready queue.

In this, the process that comes first will be executed first and the next process starts only after the previous gets fully executed.

Here we are considering that arrival time for all processes is 0.

What is Waiting Time and Turnaround Time?

1. Turnaround Time is the time interval between the submission of a process and its completion.

Turnaround Time = completion of a process – submission of a process

2. Waiting Time is the difference between turnaround time and burst time

Waiting Time = turnaround time – burst time

we have assumed arrival times as 0, so turn around and completion times are same

FCFS (Example)

Process	Duration	Oder	Arrival Time
P1	24	1	0
P2	3	2	0
P3	4	3	0

Gantt Chart :



P1 waiting time : 0

P2 waiting time : 24

P3 waiting time : 27

The Average waiting time :

$$(0+24+27)/3 = 17$$

Code:-

```
#include<iostream>
using namespace std;
int main(){
    int i,n;
    cout<<"Enter total Number of Processes: ";
    cin>>n;
    int bt[n],wt[n],ct[n];
    for(i=0;i<n;i++){
        cout<<"Enter Burst Time for Process P"<<i<<": ";
        cin>>bt[i]; }
    wt[0]=0;
    for(i=1;i<=n;i++){
        wt[i]=wt[i-1]+bt[i-1]; }
    for(i=0;i<n;i++){
        ct[i]=wt[i]+bt[i];
    }
    cout<<"Process\t Burst Time \t Waiting Time \t Completion Time \n";
    for(i=0;i<n;i++){
        cout<<"P"<<i<<"\t";
        cout<<bt[i]<<"\t\t"<<wt[i]<<"\t\t"<<ct[i]<<"\t";
        cout<<"\n"; }
    }
}
```

Output:-

```
Enter total Number of Processes: 3
Enter Burst Time for Process P0: 23
Enter Burst Time for Process P1: 3
Enter Burst Time for Process P2: 4
Process    Burst Time      Waiting Time      Completion Time
P0          23              0                  23
P1          3                23                 26
P2          4                26                 30
```

Code - (Different Arrival Time)

```
#include<iostream>
using namespace std;
int main(){
    int i,n,j;
    cout<<"Enter total Number of Processes: ";
    cin>>n;
    int bt[n],wt[n],ct[n],art[n];
    for(i=0;i<n;i++){
        cout<<"Enter Burst Time for Process P"<<i<<": ";
        cin>>bt[i];
        cout<<"Enter Arrival Time for Process P"<<i<<": ";
        cin>>art[i];
        wt[0]=0;
        for(i=1;i<=n;i++){
            int sum=0;
            for(j=0;j<i;j++){
                sum+=bt[j];
            }
            wt[i]=sum-art[i];
            ct[i]=wt[i]+bt[i];
        }
        cout<<"Process\t Burst Time \t Arrival Time \t Waiting Time \t Execution Time \n";
        for(i=0;i<n;i++){
            cout<<"P"<<i<<"\t";
            cout<<bt[i]<<"\t\t"<<art[i]<<"\t\t"<<wt[i]<<"\t\t"<<ct[i]<<"\t";
            cout<<"\n"; } }
```

Output :-

```
Enter total Number of Processes: 3
Enter Burst Time for Process P0: 23
Enter Arrival Time for Process P0: 0
Enter Burst Time for Process P1: 3
Enter Arrival Time for Process P1: 2
Enter Burst Time for Process P2: 4
Enter Arrival Time for Process P2: 3
Process  Burst Time      Arrival Time      Waiting Time      Execution Time
P0      23              0                  0                  23
P1      3               2                  21                 24
P2      4               3                  23                 27
```

Aim - Study of hardware & Software requirements of various Operating Systems**Theory -****What is an Operating System?**

An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs. For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware, although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it.

Parameters	UNIX	LINUX	Windows XP	Windows 7	Windows 10
RAM	256MB, minimum.	1 GB or greater	64 MB of RAM	1 gigabyte (GB) RAM (32-bit) or 2 GB RAM (64-bit)	1 gigabyte (GB) for 32-bit or 2 GB for 64-bit.
Processor	HP 9000/800	64-bit Opteron, EM64T	233 MHz processor	1 GHz or faster 32-bit or 64-bit processor	2 gigahertz (GHz) or faster processor or SoC.
Disk Space	Minimum of 300MB	500 MB free space	1.5 gb of free hard drive space.	16 GB available hard disk space (32-bit) or 20 GB (64-bit)	16 GB for 32-bit OS or 20 GB for 64-bit OS.
Drivers/Hardware	ODBC Driver	None	SVGA-capable video card.	DirectX 9 graphics device with WDDM 1.0 or higher driver	DirectX 9 or later with WDDM 1.0 driver

Result - Studied and Understood Hardware and Software requirements of various operating systems.



