PAPER 1 (THEORY)

(Maximum Marks: 70)

(Time allowed: Three hours)

*(Candidates are allowed additional 15 minutes for **only** reading the paper**.** They must NOT start writing during this time.)*

*Answer **all** questions in Part I (compulsory) and **six** questions from Part-II, choosing **two** questions from Section-A, **two** from Section-B and **two** from Section-C.*

All working, including rough work, should be done on the same sheet as the rest of the answer.

The intended marks for questions or parts of questions are given in brackets [ ].

PART I (20 Marks)

*Answer **all** questions.*

While answering questions in this Part, indicate briefly your working and reasoning, wherever required.

Question 1

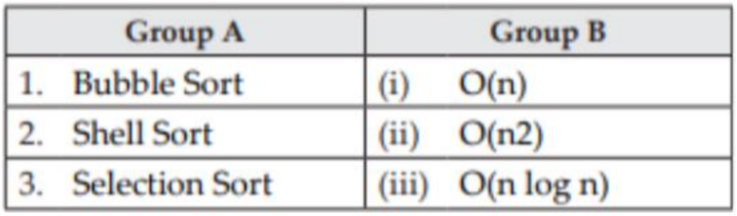
I. State the searching technique which do not require the data to be in sorted form [1]

(a) Binary Search (b) Interpolation Search c) Linear Search (d) All of the above

II. Identify the algorithm which cannot be designed without recursion [1]

(a) Tower of Hanoi (b) Fibonacci Series (c) Tree Traversal (d) None of the above

III. Match the columns: [1]



Group A	Group B
1. Bubble Sort	(i) O(n)
2. Shell Sort	(ii) O(n2)
3. Selection Sort	(iii) O(n log n)

(a) 1-(i), 2-(ii), 3-(iii) (b) 1-(ii), 2-(iii), 3-(i) (c) 1-(i), 2-(iii), 3-(ii) (d) 1-(ii), 2-(i), 3-(iii)

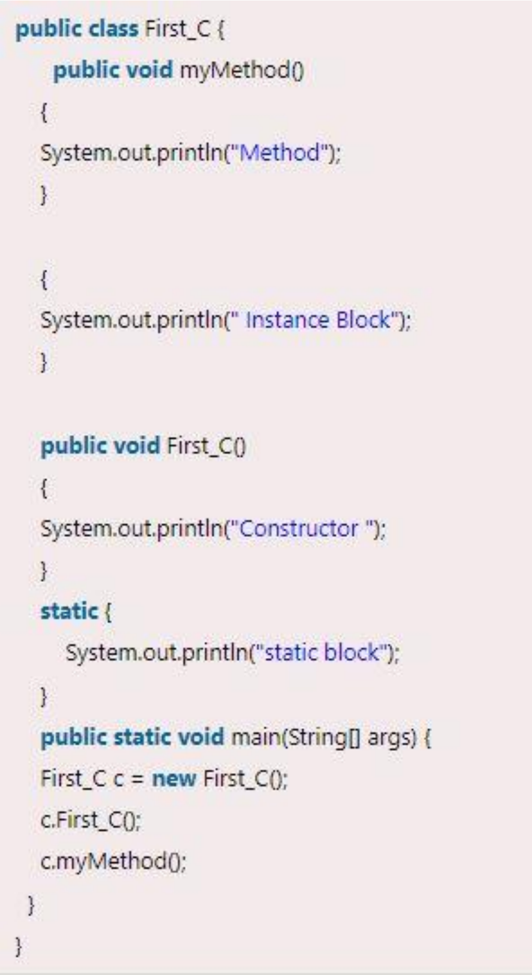
IV. push () and pop () functions are found in [1]

(a) Queues (b) lists (c) stacks (d) trees

V. ________Method used to extract a single character from a String [1]

(a) toCharArray( ) (b) getChars( ) (c) getBytes( ) (d) charAt( )

VI. What should be the execution order, if a class has a method, static block, instance block, and Constructor, as shown below? [1]



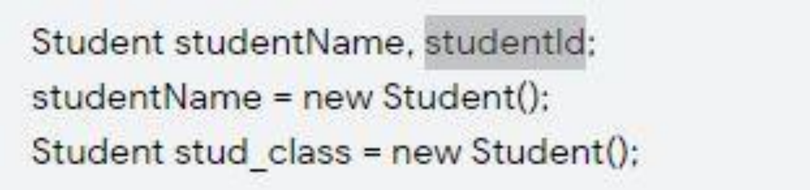
```
public class First_C {
   public void myMethod()
  {
  System.out.println("Method");
  }

  {
  System.out.println(" Instance Block");
  }

  public void First_C()
  {
  System.out.println("Constructor ");
  }
  static {
    System.out.println("static block");
  }
  public static void main(String[] args) {
  First_C c = new First_C();
  c.First_C();
  c.myMethod();
 }
}
```

(a) Instance block, method, static block, and constructor
(b) Method, constructor, instance block, and static block
(c) Static block, method, instance block, and constructor
(d) Static block, instance block, constructor, and method

VII. In which memory a String is stored, when we create a string using new operator? [1]

(a) Stack (b) String memory (c) Heap memory (d) Random storage space

VIII. Given that Student is a class, how many reference variables and objects are created by the following code? [1]



```
Student studentName, studentId;
studentName = new Student();
Student stud_class = new Student();
```

(a) Three reference variables and two objects are created.
(b) Two reference variables and two objects are created
(c) One reference variable and two objects are created.
(d) Three reference variables and three objects are created.

IX. A “**this**” operator refers to ___ variable. [1]

(a) Global variable (b) Method local variable (c) Instance variable d) class variable.

X. Why tree is recursive data structure? [1]

Question 2

(i) Convert the following infix notation to postfix form. [2]

(P / Q – R) * (S + T)

(ii) A square matrix A[m*m] is stored in the memory with each element requiring 2 bytes [2]

of storage. If the base address A[1] [1] is 1098 and the address at A [4] [5] is 1144,

determine the order of the matrix A[m × m] when the matrix is stored Column Major wise.

(iii) The following function **quiz**( ) is a part of some class. Assume ‘n’ is a positive integer, greater than 0. Answer the given questions along with **dry run / working**.

```
int quiz( int n)
{
    if ( n <= 1 )
    return n;
    else
    return (--n % 2) + quiz(n/10);
}
```

(a) What will the function **quiz (**) return when the value of **n=36922**? [2]

(b) State in one line what does the function **quiz ()** do, apart from recursion? [1]

iv) (a) A car dealership needs a program to store information about the cars for sale. [1]

For each car, they want to keep track of the following information: number of doors
(2 or 4), whether the car has air conditioning, and its average number of miles per gallon.
Which of the following is the best design?

I. Use four unrelated classes: Car, Doors, AirConditioning, and MilesPerGallon
II. Use a class Car with three subclasses: Doors, AirConditioning, and MilesPerGallon
III. Use a class Car, with fields: numDoors, hasAir, and milesPerGallon

IV. Use a class Car, with subclasses of Doors, AirConditioning, and MilesPerGallon.

V. Use classes: Doors, AirConditioning, and MilesPerGallon, each with a subclass Car

The following **reverseString () is part of String_Reverse class. Assume “str” is a valid string.** Answer the given questions along with dry run / working.

```
class String_Reverse

{

  void reverseString (String str)

  {

        if ((str==null) || (str. length () <= 1))

        System.out.println(str);

    else

    {

      System.out.println(str. charAt (str. length ()-1));

      reverseString (str. substring (0, str. length ()-1));

    }

  }

}
```

(b) What will the function reverseString ( ) return when the value of **str = "SoftwareTestingHelp".** [2]

PART II – 50 MARKS

*Answer six questions in this part, choosing **two** questions from Section A, **two** from Section B and **two** from Section C.*

SECTION - A

*Answer **any two** questions.*

Question 3

(i) (a) Write the output for the following: [2]

```
String s1 = "phoenix";
String s2 =‘‘island’’;
System.out.println (s1.substring (0).concat (s2.substring (2)));
System.out.println(s2.toUpperCase());
```

(b) What is the difference between the linear search and the binary search technique? [2]

(c) State any one purpose of using the keyword super in Java programming. [1]

(ii) The following functions numbers (int) and numbers1 (int) are a part of some class. Answer the questions given below showing the dry run/working:

```
public void numbers (int n)
{
if (n > 0)
{
System.out. print(n + " " );
numbers (n-2);
System.out.print(n + " ");
}
}
public String numbers1 (int n)
{
if (n < = 0)
return " ";
return numbersl(n-1) + n + " ";
}
```

(a) What will be the output of the function numbers (int n) when n = 5? [2]

(b) What will the function numbersl (int n) return when n = 6? [2]

(c) State in one line what is the function numbersl (int) doing apart from recursion? [1]

Question 4

(i) The following function **witty()** is a part of some class. What will be the output of the function witty( ) when the value of **n** is ‘**SCIENCE’** and the value of **p** is **5**. Show the dry run/working: [5]

```
void witty(String n,int p)
{
if (p < 0)
System.out.println(" ");
else
{
System.out.println(n.charAt(p) + " . ");
witty (n, p - 1);
System.out.print(n.charAt(p));
}
}
```

(ii) Convert the following infix notation to its postfix form: [3]

A + (B + C) + (D + E) * F/G

(iii) Differentiate between the keywords extends and implements. [2]

Question 5

(i) Write the algorithm for push operation (to add elements) in an array based stack. [2]

(ii) State two advantages of using the concept of inheritance in Java. [2]

(iii) The disadvantage of the array is? [1]

(a) Stack and Queue data structures can be implemented through an array.

(b) Index of the first element in an array can be negative

(c) Wastage of memory if the elements inserted in an array are lesser than the allocated size

(d) Elements can be accessed sequentially.

(iv) The following functions are part of some class:

```
void fun1(char s[ ],int x)

{

System.out.println(s);

char temp;

if(x<s.length/2)

{

temp=s[x];

s[x]=s[s.length-x-1];

s[s.length-x-1 ]=temp;

fun1(s, x+1);

}

}

void fun2(String n)
```

```
{
char c[ ]=new char[n.length()];
for(int i=0;i<c.length; i++)
c[i]=n.charAt(i);
fun1(c,0);
}
```

(a) What will be the output of **fun1()** when the value of **s[ ]={'J','U','N','E'} and x = 1?** [2]

(b) What will be the output of **fun2( )** when the value of **n = 'SCROLL"?** [2]

(c) State in one line what does the function **fun1()** do apart from recursion. [1]

SECTION – B

*Answer **any two** questions.*

Each program should be written in such a way that it clearly depicts the logic of the problem.

This can be achieved by using mnemonic names and comments in the program.

(Flowcharts and Algorithms are not required.)

The programs must be written in Java.

Question 6

Design a class **Pronic** to check if a given number is a pronic number or not. [A number is said to be pronic if the product of two consecutive numbers is equal to the number] [10]

Example: 0 = 0 × 1

2 = 1 × 2

6 = 2 × 3

12 = 3 × 4

thus, 0, 2, 6, 12... are pronic numbers.

Some of the members of the class are given below:

Class name : **Pronic**

Data members/instance variables :

num : to store a positive integer number

Methods / Member functions:

Pronic( ) : default constructor to initialize the data member with legal initial value

void acceptnum( ) : to accept a positive integer number

boolean ispronic(int v) : returns *true* if the number *'num'* is a *pronic* number, otherwise returns false using **recursive technique**

void check( ) : checks whether the given number is a pronic number by invoking the function ispronic() and displays the result with an appropriate message.

Specify the class **Pronic** giving details of the **constructor( ), void acceptnum( ), boolean ispronic(int)** and **void check( )**. Define a **main( )** function to create an object and call the functions accordingly to enable the task.

Question 7

Design a class **OddEven** to arrange two single dimensional arrays into one single dimensional array, such that the odd numbers from both the arrays are at the beginning followed by the even numbers. [10]

Example:

Array 1: { 2, 13, 6, 19, 26, 11, 4 }

Array 2: { 7, 22, 4, 17, 12, 45 }

Arranged Array = { 13, 19 11, 7, 17, 45, 2, 6, 26, 4, 22, 4, 12 }

Some of the members of the class are given below:

Class name **: OddEven**

Data members/instance variables:

a[ ] : to store integers in the array

m : integer to store the size of the array

Methods / Member functions:

OddEven(int mm) : parameterised constructor to initialize the data member m=mm

void fillarray( ) : to enter integer elements in the array

OddEven arrange(OddEven P, OddEven Q) : stores the odd numbers from both the parameterized object arrays followed by the even numbers from both the arrays and returns the object with the arranged array

void display( ) : displays the elements of the arranged array

Specify the class **OddEven** giving details of **the constructor( ), void fillarray( ), OddEven arrange(OddEven, OddEven) and void display( ).** Define a **main**( ) function to create objects and call the functions accordingly to enable the task.

Question 8 [10]

A class **Encrypt** has been defined to replace only the vowels in a word by the next corresponding vowel and forms a new word. i.e. A → E, E → I, I → O, O → U and U → A

Example:

Input: COMPUTER

Output: CUMPATIR

Some of the members of the class are given below:

Class name	**: Encrypt**
Data members/instance variables	**:**
wrd	: to store a word
len	: integer to store the length of the word
newwrd	: to store the encrypted word
Methods / Member functions:	
Encrypt( )	: default constructor to initialize data members with legal initial values
void acceptword( )	: to accept a word in UPPER CASE
void freqvowcon( )	: finds the frequency of the vowels and consonants in the word stored in 'wrd' and displays them with an appropriate message
void nextVowel( )	: replaces only the vowels from the word stored in 'wrd' by the next corresponding vowel and assigns it to 'newwrd', with the remaining alphabets unchanged
void disp( )	: Displays the original word along with the encrypted word

Specify the class **Encrypt** giving details of the **constructor ( ), void acceptword( ), void freqvowcon( ), void nextVowel( ) and void disp( )**. Define a **main** ( ) function to create an object and call the functions accordingly to enable the task.

SECTION – C

*Answer any **two** questions.*

Each program should be written in such a way that it clearly depicts the logic of the problem stepwise.

This can be achieved by using comments in the program and mnemonic names or pseudo codes for algorithms.

The programs must be written in Java and the algorithms must be written in general / standard form, wherever required / specified.

*(Flowcharts are **not** required.)*

Question 9

Holder is a kind of data structure which can store elements with the restriction that an element can be added from the rear end and removed from the front end only. The details of the class **Holder** is given below:

Class name : Holder

Data members/instance variables :

Q[ ] : array to hold integers

cap : maximum capacity of the holder

front : to point the index of the front end

rear : to point the index of the rear end

Methods / Member functions:

Holder(int n) : constructor to initialize cap=n, front= 0 and rear=0

void addint(int v) : to add integers in the holder at the rear end if possible, otherwise display the message **" HOLDER IS FULL "**

int removeint( ) : removes and returns the integers from the front end of the holder if any, else returns −**999**

void show( ) : displays the elements of the holder

(i) Specify the class **Holder** giving details of the functions **void addint(int) and int removeint( ).** Assume that the other functions have been defined. The **main** ( ) function and algorithm need **NOT** be written [4]

(ii) Name the entity described above and state its principle. [1]

Question 10 [5]

A super class **Bank** has been defined to store the details of the customer in a bank. Define a subclass **Interest** to calculate the compound interest. The details of the members of both the classes are given below:

Class name : Bank Data

Members/instance variables:

name : to store the name of the customer acc_no : integer to store the account number

principal : to store the principal amount in decimals

Methods / Member functions:

Bank(...) : parameterized constructor to assign values to the data members

void display( ) : to display the customer details

Class name Interest

Data members/instance variables:

rate : to store the interest rate in decimals

time : to store the time period in decimals

Methods / Member functions:

Interest(...) : parameterized constructor to assign values to the data members of both the classes

double calculate( ) : to calculate and return the compound interest using the formula [CI = P (1 + R/100)N − P]

where, P is the principal, R is the rate and N is the time

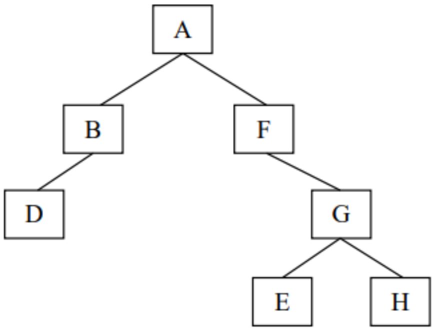
void display( ) : to display the customer details along with the compound interest

Assume that the super class **Bank** has been defined. Using the concept of **inheritance**, specify the class **Interest** giving the details of **the constructor(...), double calculate( ) and void display( ).** The **super class**, **main function and algorithm need NOT be written.**

Question 11

(i) State one advantage and one disadvantage of using recursion over iteration. [2]

(ii) Answer the following questions from the diagram of a Binary Tree given below:



(a) Write the in-order traversal of the above tree structure. [1]

(b) Name the children of the nodes B and G. [1]

(c) State the root of the right sub tree. [1]