ICSE COMPUTER APPLICATIONS 2026
SOLUTION PAPER

SECTION A

Question 1

Choose the correct answers to the questions from the given options.

(Do not copy the questions, write only the correct answers.)

(i) The full form of *JVM* is:

(a) Java Visible Machine

(b) Java Virtual Mode

(c) Java Virtual Machine

(d) Java Visible Mode

Option c. Java Virtual Machine

Reason: JVM is the runtime environment for Java bytecode.

(ii) Which of the following occupies *2 bytes* of storage?

(a) 25

(b) AM

(c) 35.2

(d) \\

Option d. \\

Reason: In Java, char occupies **2 bytes** → '\\' is actually the character '\' .

(iii) In a statement $c = c + (x * d + e);$ which ***variable*** is an ***accumulator***?

(a) *d*

(b) *c*

(c) *e*

(d) *x*

Option b. c

Reason: c = c + (x*d+e), here the variable c accumulates the value of the expression

(iv) Which of the following is ***NOT*** an access specifier?

(a) private

(b) protected

(c) package

(d) public

Option c. package
Reason: package is **not** an access specifier; it is a keyword used to declare a package.

(v) What is the output of the statement *Math.pow (36, 6/5);* ?

(a) 36.0

(b) 1.0

(c) 73.71

(d) 6.0

Option a. 36.0
Reason: 6/5 = 1 (As integer division in Java results in an integer value) So, Math.pow(36,1) = 36.0

(vi) Read the *if program segment* given below:

```
if (a > b)
z = 25;
else
z = 35;
```

Which one of the following is the *correct conversion* of the *if program segment* to *ternary*?

(a) z = a > b ? 35 : 25;

(b) z = a > b ? 25 : 35;

(c) z = a > b : 35 ? 25;

(d) z = a > b : 25 ? 35;

Option b. z = a > b ? 25 : 35;
Reason: Syntax of ternary operator is Condition ? True value : False value

(vii) The output of the statement:

System.out.println*(Character.toUpperCase('b') + 2);* is:

(a) 66

(b) 100

(c) 68

(d) 98

Option c. 68
Reason: 'B'+2 = 66+2= 68

(viii) Consider the following statements:

```
Computer desktop = new Computer( );
Computer Mainframe = new Computer( );
```

Name the ***objects*** of the class given above:

(a) Desktop, Mainframe

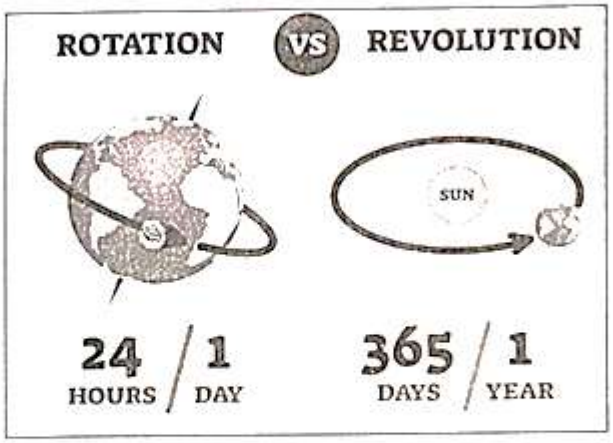
(b) desktop, Mainframe

(c) Computer, Mainframe

(d) Computer, desktop

Option b. desktop, Mainframe
Reason: Objects are variable names → desktop & Mainframe

(ix) The earth spins on its axis completing one rotation in a day. The earth revolves around the sun in 365 days to complete one revolution. What is the ***Java concept*** depicted in the given picture?



(a) Array

(b) Condition

(c) Nested loop

(d) While loop

Option c. Nested loop
Reason: Rotation inside revolution → loop inside loop analogy

(x) In the following method prototype to accept a character, an integer and return YES or NO, fill in the blank to complete the method prototype.

public _____ *someMethod (char ch, int n)*

(a) boolean

(b) String

(c) int

(d) double

Option b. String

Reason: Returns YES/NO → String type and not Boolean as Boolean values are only true and false.

(xi) In a calculator which Java feature allows ***multiple methods named calculate()*** for the different operations?

(a) abstraction

(b) inheritance

(c) encapsulation

(d) polymorphism

Option d. polymorphism

Reason: In polymorphism, we have function overloading, which states that we can have methods with the same name and different forms in a class.

(xii) **Assertion (A):** The result of the Java expression *3 + 7/2* is *6*.

Reason (R): According to the hierarchy of operators in Java, addition is done first followed by division.

(a) (A) is true and (R) is false.

(b) (A) is false and (R) is true.

(c) Both (A) and (R) are true and (R) is the correct explanation of (A).

(d) Both (A) and (R) are true, but (R) is not the correct explanation of (A).

Option a. (A) true, (R) false

Reason: As per the operator precedence rule, division happens before addition.

(xiii) What is the ***type of parameter*** to be given for the method ***parseInt()?***

(a) double

(b) String

(c) char

(d) int

Option b. String

Reason: parseInt() accepts numbers in String format

(xiv) To extract the word ***NOW*** from the word ***"ACKNOWLEDGEMENT"***, the Java statement ***"ACKNOWLEDGEMENT".substring(3, _____)*** is used.

Choose the correct number to fill in the blank.

(a) 6

(b) 7

(c) 5

(d) 8

Option a. 6
Working:
"ACKNOWLEDGEMENT"
Indexes: N = 3 and W = 5 → substring(3,6)

(xv) String a[ ] = {"Atasi", "Aditi", "Anant", "Amit", "Ahana"};
System.out.println(a[1].charAt(1) + "*" + a[2].charAt(2));
The ***output*** of the above statement is:

(a) da

(b) d * a

(c) ti

(d) t * i

Option b. d*a
Working:
a[1] = "Aditi" → charAt(1) = d
a[2] = "Anant" → charAt(2) = a

(xvi) Which of the following ***String*** methods ***returns a negative value***?

(a) length()

(b) equals()

(c) compareTo()

(d) charAt()

Option c. compareTo()
Reason: Returns a negative, zero, or positive value based on whether the first string is smaller or equals or greater than the second string.

(xvii) An array with 3 elements is arranged in ascending order as follows:

Name the technique used:

(a) Bubble sort

(b) Linear Search

(c) Selection sort

(d) Binary Search

Option a. Bubble sort
Reason: Adjacent elements swapping is shown in the diagram

(xviii) The sales made by *5* salesmen selling *5* products is stored in a two-dimensional array of ***integer data type***. How many ***bytes*** does the array occupy?

(a) 25

(b) 200

(c) 50

(d) 100

Option d. 100 bytes
Working: 5 × 5 = 25 elements
int = 4 bytes → 25 × 4 = 100 bytes

(xix) **Assertion (A):** The substring() method modifies the original String.

Reason (R): The substring() method can extract part of a String starting from a specific index.

(a) (A) is true and (R) is false.

(b) (A) is false and (R) is true.

(c) Both (A) and (R) are true and (R) is the correct explanation of (A).

(d) Both (A) and (R) are true, but (R) is not the correct explanation of (A).

Option b.
Reason: substring does NOT modify original string → A is false, but R is true

(xx) In *constructor overloading* all constructors should have the same name as of the class but with a different set of _____.

(a) Access specifiers

(b) Classes

(c) Return type

(d) Parameters

Option d. Parameters

Reason: Constructor overloading depends on the parameters passed to the constructor

Question 2.

(i) *Rewrite* the following program segment using a *for loop*.

```
int a = 5, b = 10;
while (b > 0)
{ b -= 2;
}
System.out.println (a * b);
```

Ans. Convert while to a for loop

```
int a = 5, b;
for(b=10; b > 0; b -= 2){ }
System.out.println(a + b);
```

(ii) *Evaluate* the Java expression:

$x = a * b \% (++c) + (++a) + (--b);$

if $a = 7,\ b = 8,\ c = 2$

Ans. Evaluate the expression

x = a * b % (++ c) + (++a) + (--b)

Given, a = 7, b = 8, c = 2

Working:

- ++c => c = 3
- ++a => a = 8
- --b => b=7

Final solution:

x = 8 * 7 % 3 + 8 + 7

= 56 % 3 + 8 + 7

= 2 + 8 + 7

Answer: **17**

(iii) Write the *Java expression* to find the *sum of cube root of x and the absolute value of y*.

Ans. Expression

```
double z=Math.cbrt(x) + Math.abs(y);
```

(iv) Users must be above 10 years to open a self-operated bank account. Write this logic using a ***ternary operator*** and store the result (the eligibility message) in a String variable named idStatus and print it.

Ans. Ternary operator

```
String idStatus = (age > 10) ? "Eligible" : "Not Eligible";
System.out.println(idStatus);
```

(v) Give the output of the following program segment:

```
String S = "GRACIOUS".substring(4);
System.out.println(S);
System.out.println ("GLAMOROUS".endsWith(S));
```

Ans. Output

IOUS
false

Working:
G R A C I O U S
0 1 2 3 4 5 6 7
S = "IOUS"
"GLAMOROUS".endsWith("IOUS") → false

(vi) Give the ***output*** of the following program segment and mention ***how many times*** the loop is executed.

```
int K = 1;
do
{ K += 2;
System.out.println (K);
} while (K <= 6);
```

Ans. Loop execution

Iterations:

- K=1 → K+=2 → 3
- K=3 → K+=2 → 5
- K=5 → K+=2 → 7 (stops)

Output:
3
5
7
Loop runs **3 times**

(vii) The following program segment ***calculates and displays the factorial of a number.*** [Example : Factorial of 5 is 1 x 2 x 3 x 4 x 5 = 120]

```
int p, n = 5, f = 0;
for (p = n; p > 0; p – –)
f * = p;
System.out.println(f);
```

Name the type of error if any; correct the statement to get the desired output.

Ans. Error correction

Error: **Logical error**

Correction:

```
int p, n = 5, f = 1;
for(p = n; p > 0; p--)
   f *= p;
System.out.println(f);
```

(viii) ***int X[ ][ ] = {{4, 5}, {7, 2}, {19, 4}, {7, 4}};***

Write the ***index of*** the ***maximum element*** and the ***index of*** the ***minimum element*** of the array.

Ans. Index

Array: {4,5}, {7,2}, {19,4}, {7,4}
Max = 19 → index (2,0)
Min = 2 → index (1,1)

(ix) The following program segment ***swaps the first element and the second element*** of the given array ***without using the third variable***, fill in the blanks with appropriate java statements:

```
void swap()
{ int x[ ] = {4, 8, 19,24, 15};
(1) _____;
(2) _____;
x[0] = x[0] / x[1];
System.out.println(x[0]+" "+x[1]);   }
```

Ans. Swap without a third variable

```
x[0] = x[0] * x[1];
x[1] = x[0] / x[1];
x[0] = x[0] / x[1];
```

(x) Name the following:

(a) The *return data type* of the method *equals()*.

(b) The *String method* which has *no parameter* and *returns a String*.

Ans.

(a) **boolean**

(b) **toUpperCase() / toLowerCase() / trim()**

SECTION B
PROGRAMS

Question 3.

Define a class named ***StepTracker*** with the following specifications:

Member Variables:

String name — stores the user's name.

int sw — stores the total number of steps walked by the user.

double cb — stores the estimated calories burned by the user.

double km — stores the estimated distance walked in kilometers.

Member Methods:

void accept() — to input the name and the steps walked ***using Scanner class methods only***.

void calculate() — calculates calories burned and distance in km based on steps walked using the following estimation table:

Metric	Calculation Formula
Calories Burned	steps walked × 0.04 (e.g., 1 step burns 0.04 calories)
Distance (Km)	steps walked / 1300 (e.g., 1300 steps is approx. 1 km)

void display() — Display the calories burned, distance in km and the user's name.

Write a main method to ***create an object*** of the class and ***invoke the methods***.

Solution.

```
import java.util.*;
class StepTracker
{
   String name;
   int stp;
   double cb;
   double km;

   void accept()
   {
      Scanner sc = new Scanner(System.in);
```

```
        System.out.print("Enter name: ");
        name = sc.nextLine();
        System.out.print("Enter steps: ");
        stp = sc.nextInt();
    }

    void calculate()
    {
        cb = stp * 0.04;
        km = stp / 1300.0;
    }

    void display()
    {
        System.out.println("Name: " + name);
        System.out.println("Calories Burned: " + cb);
        System.out.println("Distance (km): " + km);
    }

    public static void main(String args[])
    {
        StepTracker obj = new StepTracker();
        obj.accept();
        obj.calculate();
        obj.display();
    }
}
```

Question 4.

Write a program to accept the designations of 100 employees in a single dimensional array. Accept the designation from the user and ***print the total number of employees*** with the designation given by the user as input.

Example:

Trainee	Manager	Chef	Manager	Director	Manager

Input: Manager ***Output: 3***

Solution.

```
import java.util.*;
class Employee
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        String emp[] = new String[100];
        for(int i=0; i<100; i++)
        {
            System.out.print("Enter designation: ");
            emp[i] = sc.nextLine();
        }
```

```
        System.out.print("Enter designation to search: ");
        String d = sc.nextLine();
        int count = 0;
        for(int i=0; i<100; i++)
        {
            if(emp[i].equalsIgnoreCase(d))
                count++;
        }
        System.out.println("Total employees: " + count);
    }
}
```

Question 5.

Write a program to accept a two-dimensional ***integer array of order 4 x 5*** as input from the user. Check if it is a ***Sparse Matrix*** or not. A matrix is considered to be a sparse, if the total number of zero elements is greater than the total number of non-zero elements. Print appropriate messages.

Example:

4	3	0	1	0
1	0	0	2	0
1	0	1	0	0
0	3	2	0	0

Number of zero elements = 11

Number of non zero elements = 9

Matrix is a Sparse Matrix

Solution.

```
import java.util.*;
class Sparse
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        int a[][] = new int[5][5];
        int zero = 0, nonZero = 0;
        System.out.println("Enter elements:");
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<5;j++)
            {
                a[i][j] = sc.nextInt();
                if(a[i][j]==0)
                    zero++;
                else
                    nonZero++;
            }
        }
        System.out.println("Zero elements: " + zero);
        System.out.println("Non-zero elements: " + nonZero);
        if(zero > nonZero)
            System.out.println("Sparse Matrix");
```

```
        else
            System.out.println("Not Sparse Matrix");
    }
}
```

Question 6.

Write a program to accept a number and check if it is a ***Mark number*** or not. A number is said to be Mark when sum of the squares of each digit is an even number as well as the last digit of the sum and the last digit of the number given is same.

Example: n = 246

sum = 2 x 2 + 4 x 4 + 6 x 6 = 56

56 is an even number as well as last digit is 6 for both sum as well as the number.

Solution.

```
import java.util.*;
class MarkNumber
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number");
        int n = sc.nextInt();
        int temp = n;
        int sum = 0;
        while(n > 0)
        {
            int d = n % 10;
            sum += d * d;
            n /= 10;
        }
        if(sum % 2 == 0 && (sum % 10 == temp % 10))
            System.out.println("Mark Number");
        else
            System.out.println("Not Mark Number");
    }
}
```

Question 7.

Define a class to *overload* the method *format* as follows:

void format ():

To print the following pattern using *Nested for loops only:*

```
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5
```

int format (String s):

To calculate and return the sum of ASCII codes of each character of the String.

Example: CAB

Output: 67 + 65 + 66

198

void format (int n):

To calculate and display the sum of natural numbers up to *n* given by the formula.

$$\frac{n(n+1)}{2}$$

Solution.

```
import java.util.*;
class Format
{
   void format()
   {
      for(int i=1; i<=5; i++)
      {
         for(int j=i; j<=5; j++)
            System.out.print(j + " ");
         System.out.println();
      }
   }

   int format(String s)
   {
      int sum = 0;
      for(int i=0; i<s.length(); i++)
         sum += (int)s.charAt(i);
      return sum;
   }

   void format(int n)
   {
      int sum = n*(n+1)/2;
      System.out.println("Sum = " + sum);
   }
}
```

Question 8.

Write a program to accept a word and print the ***Symbolic*** of the accepted word.

Symbolic word is formed by ***extracting*** the characters from the ***first consonant***, then add characters before the first consonant of the accepted word and end with "TR".

Example: AIRWAYS Symbolic word is RWAYSAITR

BEAUTY Symbolic word is BEAUTYTR

Solution.

```
import java.util.*;
class Symbolic
{
   public static void main(String args[])
   {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter word: ");
      String s = sc.next().toUpperCase();
      String vowels = "AEIOU";
      int pos = -1;
      for(int i=0;i<s.length();i++)
      {
         if(vowels.indexOf(s.charAt(i)) == -1)
         {
            pos = i;
            break;
         }
      }
      String result;
      if(pos == -1)
         result = s + "TR";
      else
         result = s.substring(pos) + s.substring(0,pos) + "TR";
      System.out.println("Symbolic word: " + result);
   }
}
```
