ICSE Class 10 Computer Applications Specimen Paper 2

Written by the Future Topper subject team on the CISCE examination pattern. Not an official CISCE paper.
Found a mistake? Write to contactfuturetopper@gmail.com

Maximum Marks: 100 | Time Allowed: Two hours | Reading Time: 15 minutes

Section A is compulsory. Attempt any four questions from Section B.

General Instructions

Write your answers on the separate answer paper provided. The first 15 minutes are for reading the paper, and you may not write during that time. Marks for each question, or part of a question, are shown in brackets. Section A carries 40 marks and must be attempted in full. Section B carries 60 marks, and you must answer any four of the seven questions in it. The programming language is Java. Write your programs in Java and give meaningful names to the variables you use. A short description of each variable, or a comment inside the program, will be credited. Indent your code so that the structure of every loop and every block is clear. Flow charts and algorithms are not required unless a question asks for one.

SECTION A (40 Marks)

Attempt all questions from this Section.

Question 1 [20]

Choose the correct answer for each part. Write only the option you choose.

(i) The principle by which the data of a class is kept hidden from the outside world and is reached only through the methods of that class is called:

(a) inheritance (b) polymorphism

(c) encapsulation (d) abstraction

(ii) The range of values that can be stored in a variable of type byte is:

(a) -128 to 127 (b) 0 to 255

(c) -127 to 128 (d) -32768 to 32767

(iii) The value printed by System.out.println(Math.round(7.5) + Math.round(-7.5)); is:

(a) 0 (b) 16 (c) -1 (d) 1

(iv) The operator that is evaluated last in the expression a + b * c > d is:

(a) + (b) * (c) > (d) they are evaluated left to right

(v) The loop that always executes its body at least once is:

(a) the for loop (b) the while loop

(c) the do while loop (d) none of these

(vi) The statement that is used to leave a switch block once a matching case has been dealt with is:

(a) continue (b) return (c) exit (d) break

(vii) The value of the expression "12" + 3 + 4 in Java is:

(a) 1234 (b) 19 (c) 127 (d) 1270

(viii) The value returned by "Specimen".charAt(4) is:

(a) c (b) i (c) m (d) e

(ix) The method that removes the blank spaces present at both ends of a string is:

(a) trim() (b) concat() (c) replace() (d) startsWith()

(x) The value returned by "Report".compareTo("Reply") is:

(a) 0 (b) a positive value (c) a negative value (d) 1

(xi) The extraction of a primitive value out of an object of its wrapper class is called:

(a) autoboxing (b) promotion (c) parsing (d) unboxing

(xii) The number of comparisons needed in the worst case to find an element by binary search in a sorted array of 16 elements is:

(a) 16 (b) 8 (c) 5 (d) 4

(xiii) In the selection sort technique the position of the smallest remaining element is found and that element is then:

(a) deleted from the array (b) exchanged with the element at the current position

(c) copied into a new array (d) compared with its neighbour only

(xiv) A member of a class that is shared by every object of that class, there being only one copy of it in memory, is declared using the keyword:

(a) final (b) private (c) static (d) void

(xv) The return type that a constructor is written with is:

(a) void (b) the name of the class

(c) int (d) a constructor has no return type at all

(xvi) The keyword that is used inside a method to send a value back to the statement that called it is:

(a) break (b) new (c) static (d) return

(xvii) When an array is passed to a method, a change made to an element of the array inside that method:

(a) is lost when the method ends (b) is seen in the original array as well

(c) causes a compilation error (d) creates a second array

(xviii) A fraudulent email that appears to come from a bank and asks the reader to confirm a password on a lookalike page is an instance of:

(a) phishing (b) a firewall (c) encryption (d) a software patch

(xix) The branch of computing concerned with machines that are designed, built and programmed to carry out physical tasks in place of a person is:

(a) robotics (b) blockchain (c) cloud computing (d) 5G

(xx) Renting storage and computing power from a provider over the internet, instead of buying and housing the servers yourself, is known as:

(a) the Internet of Things (b) cloud computing

(c) artificial intelligence (d) cyber hygiene

Question 2 [20]

(i) Give the output of the following fragment. **[2]**

```
System.out.println(Math.max(-9, -15));
System.out.println(Math.floor(-2.7));
```

(ii) State the difference between an object and a class, and write a single statement that creates an object named b1 of a class named Book using a constructor that takes no argument. **[2]**

(iii) Give the output of the following fragment. **[2]**

```
String a = "Specimen";
System.out.println(a.toUpperCase().charAt(3));
System.out.println(a.endsWith("men"));
```

(iv) Convert the following if else statement into a single statement that uses the conditional operator. **[2]**

```
if (n > 0)
   sign = 1;
else
   sign = -1;
```

(v) Explain, with one example of each, the difference between unary and binary operators. **[2]**

(vi) Give the output of the following fragment. **[2]**

```
int k = 4;
System.out.println(k++ + " " + k);
System.out.println(++k + " " + k);
```

(vii) Name the two methods of the Character class that would be used to test whether a character is a digit and whether a character is a blank space. **[2]**

(viii) Give the output of the following fragment. **[2]**

```
Integer n = 45;
int p = n;
System.out.println(p + 5);
System.out.println(Integer.toString(p).length());
```

(ix) Write the Java statement that declares an array named marks able to hold the marks of 12 students, each mark being a whole number, and state the index of the seventh element of that array. **[2]**

(x) Name any two tasks that artificial intelligence is commonly put to, and state one risk that comes with handing such a task over to a machine. **[2]**

SECTION B (60 Marks)

Attempt any four questions from this Section. Each question in this Section carries 15 marks.

Question 3 [15]

(a) Define a class named Bill as described below. **[15]**

Class name: Bill

Data members: units of type int to hold the number of electricity units consumed, and charge and total both of type double

Bill(int u): a parameterised constructor to assign the number of units consumed

void compute(): to work out the charge at the rate of 4 rupees a unit for the first 100 units, 6 rupees a unit for the next 200 units and 8 rupees a unit for every unit above 300, and then to add a fixed meter rent of 50 rupees to the charge to give the total payable

void print(): to print the units consumed, the charge and the total payable, each on a line of its own

Write a main method to create an object for a consumer who has used 450 units, call the two methods through that object, and state the output the program produces.

Question 4 [15]

Give the output of each of the following fragments. Show the working wherever a value changes more than once.

(a) Give the output and state the value of x when the loop ends. **[5]**

```
int x = 1, y = 20;
while (x < y)
{
    System.out.print(x + " ");
    x = x * 2;
    y = y - 2;
}
```

(b) Give the output of the following fragment. **[5]**

```
String t = "Learn Java Today";
int c = 0;
```

```
for (int i = 0; i < t.length(); i++)
{
   if (Character.isUpperCase(t.charAt(i)))
      c++;
}
System.out.println(c);
System.out.println(t.indexOf("Java"));
System.out.println(t.substring(6).trim().length());
```

(c) Give the output of the following fragment. **[5]**

```
for (int i = 1; i <= 4; i++)
{
   for (int j = 1; j <= i; j++)
   {
      System.out.print(i * j + " ");
   }
   System.out.println();
}
```

Question 5 **[15]**

(a) Write a program to store the names of eight cities in a single dimensional array of strings. The program should then accept a name typed by the user and search for it in the array using the linear search technique. If the name is found the program should print the position at which it was found, counting the first city as position one. If it is not found the program should say that the city is not in the list. **[15]**

Question 6 **[15]**

(a) Write a program to accept a sentence in mixed case and to print the number of words it contains, the number of vowels it contains, and the first letter of every word on a single line. Assume that the words of the sentence are separated by exactly one blank space and that the sentence does not end with a full stop. **[15]**

Question 7 **[15]**

(a) The following program has been written to print the factorial of a number stored in the variable n. It contains four errors. Rewrite the whole program after correcting the errors, underline each correction you make, and state the output of the corrected program. **[8]**

```
class Factorial
{
   public static void main(String args[])
   {
      int n = 5;
      int f = 0;
      for (int i = 1; i =< n; i++)
      {
         f = f * i;
      }
      System.out.println("Factorial = " + F);
   }
```

(b) Rewrite the following fragment using a while loop in place of the for loop, without changing the output, and state the output. **[7]**

```
int t = 0;
for (int i = 1; i <= 9; i = i + 2)
{
   t = t + i * i;
}
System.out.println("t = " + t);
```

Question 8 **[15]**

(a) Write a program using a user defined method named int sumDigits(int n) that accepts a whole number, adds up its digits and returns that sum. The main method should read a number from the user, pass it to the method, and print both the number and the sum of its digits. **[8]**

(b) Write a program to print the following pattern exactly as it is shown, using a loop inside a loop. **[7]**

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

Question 9 **[15]**

(a) Define a class named Student as described below. **[15]**

Class name: Student

Data members: name of type String, and m1, m2, m3, total and average, the three marks being of type int and the other two of type double

Student(): a default constructor that sets the name to a blank string and all three marks to zero

Student(String n, int a, int b, int c): a parameterised constructor that assigns the name and the three marks

void find(): to work out the total of the three marks and the average of the three marks

void show(): to print the name, the total and the average, and then to print the grade, which is A when the average is 75 or more, B when the average is 60 or more but less than 75, C when the average is 40 or more but less than 60, and D otherwise

Write a main method that creates one object using the default constructor and one object using the parameterised constructor for a student named Riya Sharma whose marks are 82, 76 and 91, calls find() and show() on the second object, and states the output the program produces.

ANSWER KEY AND MARKING SCHEME

Full code is given for every question that asked for a program. Marks shown in italics are the step marks an examiner would award. Accept any correct alternative logic that produces the same output, accept any sensible variable names, and do not deduct marks for a different but consistent style of indentation or bracing. For an output question the printed text must match exactly, including spaces.

Question 1 [20]

(i) (c) encapsulation. The data and the methods that work on it are wrapped together in one unit, and the data is reached only through those methods.

(ii) (a) -128 to 127. A byte holds 8 bits, which gives 256 different values, split about zero as -128 up to 127.

(iii) (d) 1. Math.round(7.5) gives 8 and Math.round(-7.5) gives -7, because rounding adds a half and then takes the floor, so the sum is 1.

(iv) (c) >. Multiplication is done first, then the addition, and the relational comparison is done last of the three.

(v) (c) the do while loop. Its condition is tested after the body, so the body has already run once by the time the test is reached.

(vi) (d) break.

(vii) (a) 1234. The first operand is a string, so each plus sign joins text rather than adding numbers, and the operators are worked out from left to right.

(viii) (b) i. Counting from zero, the characters of Specimen are S, p, e, c, i, m, e, n, so index 4 holds i.

(ix) (a) trim().

(x) (b) a positive value. The two strings agree for the first three characters, and at the fourth position o comes after l in the character set, so the result is positive.

(xi) (d) unboxing.

(xii) (c) 5. Each comparison halves the range, and 16 elements take at most 5 comparisons to come down to a single element and test it.

(xiii) (b) exchanged with the element at the current position.

(xiv) (c) static.

(xv) (d) a constructor has no return type at all. It is not even written as void.

(xvi) (d) return.

(xvii) (b) is seen in the original array as well. The method receives a copy of the reference, and that copy still points at the same array in memory.

(xviii) (a) phishing.

(xix) (a) robotics.

(xx) (b) cloud computing.

1 mark for each correct option. No marks are deducted for a wrong option.

Question 2 [20]

(i) -9 on the first line and -3.0 on the second line. Of the two negative numbers -9 is the larger. The floor of a number is the greatest whole number that is not more than it, and -3 is less than -2.7, so the answer is -3.0 and not -2.0.

-9 1. -3.0 1.

(ii) A class is a blueprint or a template that lays down what data members and what methods its members will have, and it occupies no memory for data of its own. An object is a working instance built from that blueprint, and it has its own copy of the instance variables and its own memory. The statement is Book b1 = new Book();

Difference stated 1. Correct statement creating the object 1.

(iii) C on the first line and true on the second line. The whole string first becomes SPECIMEN, and index 3 of that string holds C. The string Specimen does finish with the three letters men, so endsWith returns true.

C 1. true 1.

(iv) sign = (n > 0) ? 1 : -1;

Correct use of the conditional operator with the question mark and the colon 1. The two values given in the right order 1.

(v) A unary operator works on one operand only, as in the minus sign in -x, or the increment in k++, or the logical not in !flag. A binary operator needs two operands, one on each side of it, as in a + b, or a > b, or a % b. Any correct example of each kind is accepted.

Unary explained with an example 1. Binary explained with an example 1.

(vi) 4 5 on the first line and 6 6 on the second line. The parts of each statement are worked out from left to right. In the first statement k++ hands over the old value 4 and then raises k to 5, and the second part of the same statement reads the raised value, so the line reads 4 5. In the second statement ++k raises k to 6 first and hands over 6, and the second part then reads the same 6, so the line reads 6 6.

First line 4 5 1. Second line 6 6 1.

(vii) Character.isDigit() tests whether the character passed to it is one of the figures 0 to 9. Character.isWhitespace() tests whether it is a blank space or a similar spacing character such as a tab. Character.isSpaceChar() is also accepted for the second.

isDigit() named 1. isWhitespace() or isSpaceChar() named 1.

(viii) 50 on the first line and 2 on the second line. The literal 45 is autoboxed into an Integer object, and that object is then unboxed back into the int p, so p holds 45 and p + 5 is 50. Integer.toString(45) gives the string "45", whose length is 2.

50 1. 2 1.

(ix) int marks[] = new int[12]; The seventh element is at index 6, because the indexes of a Java array begin at zero. The form int[] marks = new int[12]; is equally correct.

Declaration 1. Index 6 1.

(x) Artificial intelligence is commonly used to recognise faces and voices, to recommend films or products, to translate between languages, to read medical scans and to drive vehicles. Any two of these, or any other sensible task, are accepted. One risk is that the system learns from past data that is already biased, so it repeats that unfairness at a scale no person could, and because the reasoning inside it is hard to inspect, nobody can easily explain or challenge a decision that goes wrong.

Two tasks named 1. One genuine risk explained 1.

Question 3 [15]

(a) The completed class is as follows.

```
public class Bill
{
    int units;
    double charge, total;
    Bill(int u)
    {
        units = u;
    }
    void compute()
    {
        if (units <= 100)
            charge = units * 4;
        else if (units <= 300)
            charge = 100 * 4 + (units - 100) * 6;
        else
            charge = 100 * 4 + 200 * 6 + (units - 300) * 8;
```

```
        total = charge + 50;
    }
    void print()
    {
        System.out.println("Units consumed = " + units);
        System.out.println("Charge = " + charge);
        System.out.println("Total payable = " + total);
    }
    public static void main(String args[])
    {
        Bill b = new Bill(450);
        b.compute();
        b.print();
    }
}
```

Working for 450 units: the first 100 units cost 100 x 4 = 400, the next 200 units cost 200 x 6 = 1200, and the remaining 150 units cost 150 x 8 = 1200. The charge is 400 + 1200 + 1200 = 2800, and the total is 2800 + 50 = 2850.

Output: Units consumed = 450, then Charge = 2800.0, then Total payable = 2850.0, each on a line of its own.

Note for the examiner: charge is declared as a double, so the printed values carry a decimal point even though the arithmetic is done on whole numbers. A candidate who declares charge and total as int, and so prints 2800 and 2850, is not penalised provided the class specification given in the question is otherwise followed.

Class header with the three data members correctly declared 2. Parameterised constructor 2. First slab handled 2. Second slab with only the units above 100 charged at 6 3. Third slab with only the units above 300 charged at 8 3. Meter rent of 50 added 1. print() and main() with the object created and both methods called 1. Correct output 1.

Question 4 [15]

(a) Dry run. x is 1 and y is 20. First pass: 1 is printed, x becomes 2, y becomes 18. Second pass: 2 is printed, x becomes 4, y becomes 16. Third pass: 4 is printed, x becomes 8, y becomes 14. Fourth pass: 8 is printed, x becomes 16, y becomes 12. Now 16 is not less than 12, so the loop stops.

Output on one line: 1 2 4 8

When the loop ends x holds 16.

Output 1 2 4 8 2. Four passes traced with y falling as well 2. Final value of x stated as 16 1.

(b) The sentence Learn Java Today has three capital letters, L, J and T, so c is 3. The word Java begins at index 6, counting from zero and counting the blank spaces. substring(6) gives "Java Today", there is no space at either end of it so trim() leaves it unchanged, and its length is 10.

Output: 3 on the first line, 6 on the second line, 10 on the third line.

3 2. 6 1.5. 10 1.5.

(c) The outer loop runs four times and the inner loop runs i times, printing the product i multiplied by j each time, with a blank space after every number and a new line at the end of each row.

Output, four lines in all:

1

2 4

3 6 9

4 8 12 16

First line 1. Second line 1. Third line 1. Fourth line 1. Correct number of lines with the values separated by spaces 1.

Question 5 [15]

(a) The program is as follows.

```
import java.util.Scanner;
public class CitySearch
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        String city[] = {"Agra", "Bhopal", "Chennai", "Delhi", "Indore", "Jaipur", "Kochi", "Lucknow"};
        System.out.print("Enter the name of a city: ");
        String want = in.next();
        int pos = -1;
        for (int i = 0; i < city.length; i++)
        {
            if (city[i].equalsIgnoreCase(want))
            {
                pos = i + 1;
                break;
            }
        }
        if (pos == -1)
            System.out.println(want + " is not in the list");
        else
            System.out.println(want + " found at position " + pos);
    }
}
```

Specimen run one. Input Delhi. The output is Delhi found at position 4.

Specimen run two. Input Patna. The output is Patna is not in the list.

Note for the examiner: the strings must be compared with equals() or equalsIgnoreCase() and never with ==, which would compare references and would normally report every city as absent. Accept a solution that uses a boolean flag in place of the position variable, provided the position printed is still counted from one.

Class header, main() and the array of eight names initialised 3. Name read from the user 2. Position variable set to a value that means not found 2. Loop comparing every element with equals() or equalsIgnoreCase() 4. Position stored as i + 1 so that counting starts at one 2. Correct message in each of the two cases 2.

Question 6 [15]

(a) The program is as follows.

```
import java.util.Scanner;
public class Sentence
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String s = in.nextLine();
        int len = s.length();
```

```
        int words = 1;
        int vowels = 0;
        String first = "";
        for (int i = 0; i < len; i++)
        {
            char ch = s.charAt(i);
            if (ch == ' ')
                words++;
            char up = Character.toUpperCase(ch);
            if (up == 'A' || up == 'E' || up == 'I' || up == 'O' || up == 'U')
                vowels++;
            if (i == 0 || s.charAt(i - 1) == ' ')
                first = first + ch + " ";
        }
        System.out.println("Number of words = " + words);
        System.out.println("Number of vowels = " + vowels);
        System.out.println("First letters: " + first);
    }
}
```

Specimen run. Input Learn Java Today. There are two blank spaces, so the word count is 3. The vowels are the e and the a of Learn, the two a letters of Java and the o and the a of Today, which makes 6 in all. The first letters are L, J and T.

Output: Number of words = 3, then Number of vowels = 6, then First letters: L J T

Note for the examiner: the word counter starts at 1 and not at 0, because a sentence of one word contains no blank space at all. nextLine() must be used in place of next(), since next() would read only the first word. A candidate who tests the vowels against both cases separately, rather than converting to capitals first, is equally correct.

Class header, main() and the whole line read with nextLine() 2. Word counter started at 1 and raised at every blank space 3. Vowel test covering both capital and small letters 4. First letter of a word found by testing the position before it 4. Three required items printed 2.

Question 7 [15]

(a) The four errors are these. First, the sum variable f is set to 0, so every multiplication gives 0, and it must be set to 1. Second, the relational operator in the for statement is written as =< and must be written as <=. Third, the variable in the println is written with a capital F, while the variable declared is the small f, and Java treats the two as different names. Fourth, the closing brace of the class is missing from the end of the program.

The corrected program is as follows.

```
class Factorial
{
    public static void main(String args[])
    {
        int n = 5;
        int f = 1;
        for (int i = 1; i <= n; i++)
        {
            f = f * i;
        }
```

```
        System.out.println("Factorial = " + f);
    }
}
```

The loop multiplies 1 by 1, 2, 3, 4 and 5 in turn, so the output is Factorial = 120.

f set to 1 instead of 0 2. Operator corrected to <= 2. Capital F corrected to f 1.5. Missing closing brace of the class supplied 1.5. Output Factorial = 120 1.

(b) The while loop version is as follows.

```
int t = 0;
int i = 1;
while (i <= 9)
{
    t = t + i * i;
    i = i + 2;
}
System.out.println("t = " + t);
```

The loop adds 1, 9, 25, 49 and 81, so the output is t = 165, exactly as before.

Note for the examiner: the update of i must be written as the last statement inside the body. A candidate who places it before the addition changes the values that are squared and does not earn the mark for it.

Control variable declared and set to 1 before the loop 2. Condition i <= 9 2. Update i = i + 2 placed at the end of the body 2. Output t = 165 1.

Question 8 [15]

(a) The program is as follows.

```
import java.util.Scanner;
public class DigitSum
{
    int sumDigits(int n)
    {
        int s = 0;
        while (n > 0)
        {
            s = s + n % 10;
            n = n / 10;
        }
        return s;
    }
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = in.nextInt();
        DigitSum ob = new DigitSum();
        int ans = ob.sumDigits(num);
        System.out.println("Number = " + num);
        System.out.println("Sum of its digits = " + ans);
    }
```

```
}
```

Specimen run. Input 4729. The digits peeled off are 9, 2, 7 and 4, and their sum is 22, so the output is Number = 4729 followed by Sum of its digits = 22.

Note for the examiner: the method is not static, so an object has to be created in main() before it can be called. A candidate who declares the method as static and calls it without an object is equally correct and loses no marks. The parameter n is changed inside the method, but num in main() is untouched, because a primitive is passed by value.

Method header exactly as given in the question, returning an int 2. Running total started at 0 1. Last digit taken by the remainder on 10 and the number reduced by division by 10 2. Loop condition n > 0 1. Value returned with return 1. main() reading the number, calling the method and printing both items 1.

(b) The program is as follows.

```
public class Pattern
{
    public static void main(String args[])
    {
        for (int i = 1; i <= 5; i++)
        {
            for (int j = 1; j <= i; j++)
            {
                System.out.print(i + " ");
            }
            System.out.println();
        }
    }
}
```

The outer loop fixes the row number and the inner loop prints that same number as many times as the row number says, so the five rows read 1, then 2 2, then 3 3 3, then 4 4 4 4, then 5 5 5 5 5.

Outer loop running from 1 to 5 2. Inner loop running from 1 to i 2. The row number and not the inner counter printed 2. New line taken at the end of every row 1.

Question 9 [15]

(a) The completed class is as follows.

```
public class Student
{
    String name;
    int m1, m2, m3;
    double total, average;
    Student()
    {
        name = "";
        m1 = 0;
        m2 = 0;
        m3 = 0;
    }
    Student(String n, int a, int b, int c)
    {
        name = n;
```

```
        m1 = a;
        m2 = b;
        m3 = c;
    }
    void find()
    {
        total = m1 + m2 + m3;
        average = total / 3;
    }
    void show()
    {
        System.out.println("Name = " + name);
        System.out.println("Total = " + total);
        System.out.println("Average = " + average);
        if (average >= 75)
            System.out.println("Grade = A");
        else if (average >= 60)
            System.out.println("Grade = B");
        else if (average >= 40)
            System.out.println("Grade = C");
        else
            System.out.println("Grade = D");
    }
    public static void main(String args[])
    {
        Student s1 = new Student();
        Student s2 = new Student("Riya Sharma", 82, 76, 91);
        s2.find();
        s2.show();
    }
}
```

Working for the given marks: the total is 82 + 76 + 91 = 249, and the average is 249 divided by 3, which is 83.0. As 83 is 75 or more the grade is A.

Output: Name = Riya Sharma, then Total = 249.0, then Average = 83.0, then Grade = A, each on a line of its own.

Note for the examiner: total is declared as a double, so 249.0 is printed rather than 249. The division in find() is real division and not integer division, because total is a double, so no cast is needed there. The two constructors are legal together because they differ in their parameter lists, which is constructor overloading.

Class header with the six data members correctly declared 2. Default constructor setting the blank name and the three zero marks 2. Parameterised constructor assigning all four values 3. find() working out the total and the average 2. show() printing the name, the total and the average 2. Grade decided correctly by a chain of else if tests 3. main() creating both objects and calling the two methods on the second 1.