ICSE Class 10 Computer Applications Specimen Paper 1

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 number of bytes of memory occupied by a variable of type long is:

(a) 2 (b) 4 (c) 8 (d) 16

(ii) The value returned by Math.ceil(-4.3) is:

(a) -4.0 (b) -5.0 (c) 4.0 (d) 5.0

(iii) The wrapper class that corresponds to the primitive type char is:

(a) Char (b) Character (c) Chr (d) Characters

(iv) The value printed by System.out.println(Integer.parseInt("25") + 5); is:

(a) 255 (b) "255" (c) 20 (d) 30

(v) The string returned by "COMPUTER".substring(2, 5) is:

(a) OMP (b) MPU (c) MPUT (d) OMPU

(vi) The value printed by System.out.println((int) 'A'); is:

(a) 97 (b) A (c) 65 (d) 66

(vii) The number of times the body of the loop for (int i = 1; i <= 10; i = i + 3) is executed is:

(a) 10 (b) 3 (c) 4 (d) 11

(viii) The technique in which each pair of adjacent elements is compared and the two are exchanged if they are out of order is:

(a) selection sort (b) binary search

(c) linear search (d) bubble sort

(ix) The search technique that can be applied only to an array that is already sorted is:

(a) binary search (b) linear search

(c) sequential search (d) bubble sort

(x) For the declaration int a[] = new int[8]; the index of the last element of the array is:

(a) 8 (b) 9 (c) 0 (d) 7

(xi) The automatic conversion of a primitive value into an object of its wrapper class is called:

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

(xii) The method of the Character class that returns true when the character passed to it is a capital letter is:

(a) toUpperCase() (b) isLetter()

(c) isUpperCase() (d) equalsIgnoreCase()

(xiii) A constructor that accepts no parameter is known as:

(a) a parameterised constructor (b) a copy constructor

(c) an overloaded constructor (d) a default constructor

(xiv) Two methods of the same class having the same name but different parameter lists is an example of:

(a) method overloading (b) method overriding

(c) recursion (d) encapsulation

(xv) When a primitive value is passed to a method by call by value, a change made to the formal parameter inside the method:

(a) is passed back to the actual parameter (b) does not affect the actual parameter

(c) changes the actual parameter as well (d) causes a compilation error

(xvi) A method declared as static is normally invoked using:

(a) the object name (b) the constructor name

(c) the class name (d) the method name alone

(xvii) The value printed by System.out.println(15 % -4); is:

(a) -3 (b) 4 (c) -4 (d) 3

(xviii) The technology that keeps records in a chain of blocks spread over many computers, so that an entry once made cannot quietly be altered, is:

(a) blockchain (b) cloud computing

(c) robotics (d) 5G

(xix) Keeping a different strong password for every account and changing it from time to time is an example of:

(a) cyber crime (b) cyber hygiene

(c) phishing (d) hacking

(xx) A set of everyday objects fitted with sensors, which collect readings and exchange them over a network without a person operating them, is called:

(a) artificial intelligence (b) 5G

(c) the Internet of Things (d) cloud computing

Question 2 [20]

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

```
System.out.println(Math.sqrt(Math.abs(-16)));
System.out.println(Math.pow(2, 3));
```

(ii) State the difference between the == operator and the equals() method when they are used to compare two strings. **[2]**

(iii) Write a single Java statement that stores in a variable x of type double the value of the square root of the sum of the squares of a and b, where a and b are of type double. **[2]**

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

```
String s = "Future Topper";
System.out.println(s.indexOf('T'));
System.out.println(s.replace('o', 'a'));
```

(v) Differentiate between the isUpperCase() method and the toUpperCase() method of the Character class, giving one example of each. **[2]**

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

```
System.out.println(Integer.parseInt("12") + Double.parseDouble("3.5"));
```

(vii) What is a wrapper class? Name any two wrapper classes. **[2]**

(viii) State two differences between a static method and a non-static method of a class. **[2]**

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

```
int a = 7, b = 2;
System.out.println(a / b);
System.out.println((double) a / b);
```

(x) What is meant by a cyber law? Name the Act in force in India that deals with offences committed with the help of computers and the internet. **[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 Salary as described below. **[15]**

Class name: Salary

Data members: name of type String to hold the name of the employee, and basic, da, hra, pf and net all of type double

Salary(String n, double b): a parameterised constructor to assign the name and the basic pay

void calculate(): to work out the dearness allowance as 30 per cent of the basic pay, the house rent allowance as 15 per cent of the basic pay, the provident fund as 12 per cent of the basic pay, and the net pay as the basic pay plus the dearness allowance plus the house rent allowance minus the provident fund

void display(): to print the name, the basic pay and the net pay, each on a line of its own

Write a main method to create an object of the class for an employee named Anil Verma whose basic pay is 20000, 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 how many times the loop body is executed. **[5]**

```
int m = 5, n = 3;
for (int i = 1; i <= 3; i++)
{
    m = m + n;
    n = m - n;
    System.out.print(m + "," + n + " ");
}
```

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

```
String s = "PROGRAMMING";
int len = s.length();
for (int i = 0; i < len; i++)
{
    char ch = s.charAt(i);
```

```
    if (ch == 'R' || ch == 'M')
        System.out.print(Character.toLowerCase(ch));
    else
        System.out.print(ch);
}
System.out.println();
System.out.println(s.substring(3, 7));
System.out.println(s.lastIndexOf('M'));
```

(c) Give the values of p and q as they are printed. **[5]**

```
int a[] = {3, 7, 2, 9, 5};
int p = 0, q = 1;
for (int i = 0; i < a.length; i++)
{
    if (a[i] % 2 == 0)
        p = p + a[i];
    else
        q = q * a[i];
}
System.out.println("p = " + p);
System.out.println("q = " + q);
```

Question 5 **[15]**

(a) Write a program to store ten integers entered by the user in a single dimensional array. Arrange the numbers in ascending order using the selection sort technique only, then print the sorted array on one line, and finally print the smallest and the largest of the ten numbers. Do not use any ready made sorting method. **[15]**

Question 6 **[15]**

(a) Write a program to accept a single word typed in capital letters and to check whether it is a palindrome, that is, whether it reads the same from either end. If the word is a palindrome the program should say so. If it is not a palindrome the program should say so and then print the word in reverse order. In both cases the program should also count and print the number of vowels the word contains. **[15]**

Question 7 **[15]**

(a) The following program has been written to print the sum of the first five multiples of 3. 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 Multiples
{
    public static void main()
    {
        int sum = 0;
        for (int i = 1, i <= 5, i++)
        {
            sum = sum + i * 3
        }
```

```
        System.out.println("Sum = " sum);
    }
}
```

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

```
int i = 10;
int s = 0;
do
{
   s = s + i;
   i = i - 2;
} while (i >= 2);
System.out.println("s = " + s);
```

Question 8 **[15]**

(a) Write a program, using a loop inside a loop, to print every prime number that lies between 50 and 100 on a single line, and then to print the sum of those prime numbers. A prime number is a number that has exactly two factors, one and the number itself. **[8]**

(b) Write a program to input a two digit number and to check whether it is a special two digit number. A number is a special two digit number when the sum of its two digits added to the product of its two digits equals the number itself, as in 59, where 5 + 9 + 45 = 59. **[7]**

Question 9 **[15]**

(a) Define a class named Area that contains three overloaded methods, all named compute, as described below. **[15]**

void compute(int s): to work out and print the area of a square whose side is s, the area being s multiplied by s

void compute(int l, int b): to work out and print the area of a rectangle whose length is l and whose breadth is b

void compute(double b, double h): to work out and print the area of a triangle whose base is b and whose height is h, the area being half of the base multiplied by the height

Write a main method to create one object of the class and to invoke all three methods through it, once with the value 6, once with the values 8 and 5, and once with the values 12.0 and 5.0. State the output the program produces, and explain in one sentence how the compiler decides which of the three methods to call.

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) 8. A long is a 64 bit signed integer type, and 64 bits make 8 bytes.

(ii) (a) -4.0. The ceiling of a number is the smallest whole number that is not less than it, and -4 is greater than -4.3.

(iii) (b) Character.

(iv) (d) 30. parseInt returns the int value 25, and 25 + 5 gives 30.

(v) (b) MPU. substring(2, 5) takes the characters at the indexes 2, 3 and 4, which are M, P and U, and stops before index 5.

(vi) (c) 65. The cast converts the character to its Unicode value, and the capital letter A has the value 65.

(vii) (c) 4. The control variable takes the values 1, 4, 7 and 10, and the next value 13 fails the test.

(viii) (d) bubble sort.

(ix) (a) binary search. The method keeps halving the search range, which works only if the elements are in order.

(x) (d) 7. The array has eight elements and the indexes run from 0 to 7.

(xi) (b) autoboxing.

(xii) (c) isUpperCase().

(xiii) (d) a default constructor.

(xiv) (a) method overloading.

(xv) (b) does not affect the actual parameter. A copy of the value is handed to the method, so the original variable is untouched.

(xvi) (c) the class name.

(xvii) (d) 3. In Java the remainder takes the sign of the dividend, and the dividend 15 is positive.

(xviii) (a) blockchain.

(xix) (b) cyber hygiene.

(xx) (c) the Internet of Things.

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

Question 2 [20]

(i) 4.0 on the first line and 8.0 on the second line. Math.abs(-16) gives 16, the square root of which is 4.0, and 2 raised to the power 3 is 8.0. Both methods return a double, so both values are printed with a decimal point.

4.0 1. 8.0 1.

(ii) The == operator compares the two references and is true only when both variables point to the very same object in memory. The equals() method compares the two strings character by character and is true whenever the contents are the same, whatever objects they are held in. Two strings built in different ways can therefore be equal by equals() and still false by ==.

== compares references 1. equals() compares contents 1.

(iii) double x = Math.sqrt(a * a + b * b);

Correct use of Math.sqrt 1. Correct expression inside the brackets with the result stored in x 1.

(iv) 7 on the first line and Future Tapper on the second line. The capital T first occurs at index 7, counting from zero and counting the blank space. The only small letter o in the string is the one in Topper, so only that letter is replaced, and the original string itself is left unchanged.

7 1. Future Tapper 1.

(v) isUpperCase() takes a character and returns a boolean, true if the character is a capital letter and false otherwise, so Character.isUpperCase('K') returns true. toUpperCase() takes a character and returns a character, the capital form of the one given, so Character.toUpperCase('k') returns 'K'. The first only tests, the second converts.

isUpperCase() with its return type and an example 1. toUpperCase() with its return type and an example 1.

(vi) 15.5. parseInt returns the int 12 and parseDouble returns the double 3.5. The int is promoted to a double before the addition, so the sum is the double 15.5.

15.5 1. Reason, the int being promoted to a double 1.

(vii) A wrapper class is a class in the java.lang package that wraps a primitive value inside an object, so that the value can be treated as an object and so that the useful methods of the class, such as the parse methods, become available. Two examples are Integer and Double. Character, Long, Float and Boolean are also accepted.

Definition 1. Two correct wrapper classes named 1.

(viii) A static method belongs to the class as a whole and is called using the class name, while a non-static method belongs to an individual object and is called using an object name. A static method can be used without creating any object and can work only on static data members directly, while a non-static method needs an object to exist first and can work on the instance variables of that object.

One correct difference 1. A second correct difference 1.

(ix) 3 on the first line and 3.5 on the second line. In the first statement both operands are of type int, so integer division is carried out and the fractional part is thrown away. In the second statement the cast makes a a double, b is promoted to a double as well, and real division gives 3.5.

3 1. 3.5 1.

(x) A cyber law is the body of law that governs the use of computers, computer networks and the internet, and that lays down which acts done with them are offences, what punishment they carry, and how electronic records and digital signatures are to be treated. In India these matters are dealt with by the Information Technology Act, together with the relevant sections of the Indian Penal Code.

Meaning of cyber law 1. Information Technology Act named 1.

Question 3 [15]

(a) The completed class is as follows.

```
public class Salary
{
    String name;
    double basic, da, hra, pf, net;
    Salary(String n, double b)
    {
        name = n;
        basic = b;
    }
    void calculate()
    {
        da = basic * 30 / 100;
        hra = basic * 15 / 100;
        pf = basic * 12 / 100;
        net = basic + da + hra - pf;
    }
    void display()
    {
```

```
        System.out.println("Name = " + name);
        System.out.println("Basic pay = " + basic);
        System.out.println("Net pay = " + net);
    }
    public static void main(String args[])
    {
        Salary s = new Salary("Anil Verma", 20000);
        s.calculate();
        s.display();
    }
}
```

Working for the given data: the basic pay is 20000.0, so da = 6000.0, hra = 3000.0 and pf = 2400.0, which gives net = 20000.0 + 6000.0 + 3000.0 - 2400.0 = 26600.0.

Output: Name = Anil Verma, then Basic pay = 20000.0, then Net pay = 26600.0, each on a line of its own.

Note for the examiner: an object must be created before calculate() and display() can be called, because neither method is static. Award full marks if the candidate writes 0.30 * basic in place of basic * 30 / 100, since both give 6000.0.

Class header with the five data members correctly declared 3. Parameterised constructor assigning both values 3. calculate() with all four formulae right 4. display() printing the three required items 2. main() creating the object and calling both methods 2. Correct output 1.

Question 4 [15]

(a) Dry run. Before the loop m is 5 and n is 3. First pass: m becomes 5 + 3 = 8, then n becomes 8 - 3 = 5, and 8,5 is printed. Second pass: m becomes 8 + 5 = 13, then n becomes 13 - 5 = 8, and 13,8 is printed. Third pass: m becomes 13 + 8 = 21, then n becomes 21 - 8 = 13, and 21,13 is printed.

Output on one line: 8,5 13,8 21,13

The loop body is executed three times. Note that n is worked out using the new value of m, because the assignment to m comes first.

Values after the first pass 1. Values after the second pass 1. Values after the third pass 1. Output written in the right order with the commas and the spaces 1. Three executions stated 1.

(b) The string PROGRAMMING has 11 characters, indexed 0 to 10. Every R and every M is printed in small letters and every other character is printed unchanged, so the first line reads PrOGrAmmING. substring(3, 7) takes the characters at the indexes 3, 4, 5 and 6, which are G, R, A and M. The letter M occurs at the indexes 6 and 7, so the last occurrence is at index 7.

Output: PrOGrAmmING on the first line, GRAM on the second line, 7 on the third line.

First line PrOGrAmmING 2. GRAM 1.5. 7 1.5.

(c) Dry run over the five elements. 3 is odd, so q = 1 * 3 = 3. 7 is odd, so q = 3 * 7 = 21. 2 is even, so p = 0 + 2 = 2. 9 is odd, so q = 21 * 9 = 189. 5 is odd, so q = 189 * 5 = 945.

Output: p = 2 on the first line and q = 945 on the second line.

Recognising that only 2 is even 1. p = 2 1. Running product traced 1. q = 945 2.

Question 5 [15]

(a) The program is as follows.

```
import java.util.Scanner;
public class SortTen
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
```

```
        int a[] = new int[10];
        System.out.println("Enter ten integers");
        for (int i = 0; i < 10; i++)
            a[i] = in.nextInt();
        for (int i = 0; i < 9; i++)
        {
            int small = i;
            for (int j = i + 1; j < 10; j++)
            {
                if (a[j] < a[small])
                    small = j;
            }
            int t = a[i];
            a[i] = a[small];
            a[small] = t;
        }
        System.out.println("Numbers in ascending order");
        for (int i = 0; i < 10; i++)
            System.out.print(a[i] + " ");
        System.out.println();
        System.out.println("Smallest = " + a[0]);
        System.out.println("Largest = " + a[9]);
    }
}
```

Specimen run. For the input 45, 12, 78, 3, 56, 23, 90, 7, 34, 19 the sorted line reads 3 7 12 19 23 34 45 56 78 90, the smallest is 3 and the largest is 90.

Note for the examiner: once the array is sorted the smallest is a[0] and the largest is a[9], so no separate loop is needed. Award the last mark also to a candidate who finds the two values with a fresh loop before sorting.

Class header, main() and the array declared with ten elements 2. Input loop reading the ten numbers 2. Outer loop running from 0 to 8 with the position of the smallest remembered 3. Inner loop comparing the rest of the array with that position 3. Correct three statement swap 2. Output loop printing the sorted numbers 2. Smallest and largest taken from the first and the last positions 1.

Question 6 [15]

(a) The program is as follows.

```
import java.util.Scanner;
public class WordCheck
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a word in capital letters: ");
        String w = in.next();
        int len = w.length();
        String rev = "";
        int v = 0;
        for (int i = len - 1; i >= 0; i--)
```

```
            rev = rev + w.charAt(i);
        for (int i = 0; i < len; i++)
        {
            char ch = w.charAt(i);
            if (ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
                v++;
        }
        if (w.equals(rev))
            System.out.println(w + " is a palindrome");
        else
        {
            System.out.println(w + " is not a palindrome");
            System.out.println("Reverse = " + rev);
        }
        System.out.println("Number of vowels = " + v);
    }
}
```

Specimen run one. Input MADAM. The reverse is also MADAM, so the output is MADAM is a palindrome, followed by Number of vowels = 2.

Specimen run two. Input JAVA. The reverse is AVAJ, so the output is JAVA is not a palindrome, then Reverse = AVAJ, then Number of vowels = 2.

Note for the examiner: == must not be used to compare the two strings, since it would compare references and could report a palindrome as not being one. A candidate who uses equalsIgnoreCase() is not penalised, as the input is in capital letters.

Class header, main() and input of the word 2. Length taken and the reverse string built by a loop running backwards 4. Vowel counting loop with all five vowels tested 4. Comparison made with equals() rather than == 2. Correct message in each of the two cases, with the reverse printed only when the word is not a palindrome 2. Vowel count printed in both cases 1.

Question 7 [15]

(a) The four errors are these. First, the header of main() has no parameter, and it must be written as public static void main(String args[]). Second, the three parts of the for statement are separated by commas instead of semicolons. Third, the statement sum = sum + i * 3 has no semicolon at the end of it. Fourth, the println statement has no plus sign between the string and the variable sum.

The corrected program is as follows.

```
class Multiples
{
    public static void main(String args[])
    {
        int sum = 0;
        for (int i = 1; i <= 5; i++)
        {
            sum = sum + i * 3;
        }
        System.out.println("Sum = " + sum);
    }
}
```

The loop adds 3, 6, 9, 12 and 15, so the output is Sum = 45.

main() header corrected 2. Semicolons in the for statement 2. Missing semicolon supplied 1.5. Plus sign in the println 1.5. Output Sum = 45 1.

(b) The for loop version is as follows.

```
int s = 0;
for (int i = 10; i >= 2; i = i - 2)
{
    s = s + i;
}
System.out.println("s = " + s);
```

The loop adds 10, 8, 6, 4 and 2, so the output is s = 30, exactly as before.

Note for the examiner: a do while loop always runs its body at least once, so the conversion is safe here only because the condition is already true at the start. Accept i -= 2 for the update.

Initialisation i = 10 placed in the for statement 2. Condition i >= 2 2. Update i = i - 2 in the right place 2. Output s = 30 1.

Question 8 [15]

(a) The program is as follows.

```
public class PrimeRange
{
    public static void main(String args[])
    {
        int sum = 0;
        for (int n = 50; n <= 100; n++)
        {
            int c = 0;
            for (int d = 1; d <= n; d++)
            {
                if (n % d == 0)
                    c++;
            }
            if (c == 2)
            {
                System.out.print(n + " ");
                sum = sum + n;
            }
        }
        System.out.println();
        System.out.println("Sum = " + sum);
    }
}
```

Output: 53 59 61 67 71 73 79 83 89 97 on the first line, and Sum = 732 on the second line.

Note for the examiner: the counter c must be set to zero inside the outer loop, not before it. Accept a solution that tests divisors only up to the square root of n, or one that uses a boolean flag, provided the printed primes and the sum are right.

Outer loop covering 50 to 100 2. Factor counter reset inside the outer loop 1. Inner loop counting the factors 2. Test c == 2 to decide a prime 1. Primes printed on one line 1. Sum accumulated and printed 1.

(b) The program is as follows.

```
import java.util.Scanner;
public class SpecialNumber
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a two digit number: ");
        int n = in.nextInt();
        int t = n / 10;
        int u = n % 10;
        if (t + u + t * u == n)
            System.out.println(n + " is a special two digit number");
        else
            System.out.println(n + " is not a special two digit number");
    }
}
```

Specimen run one. Input 59. The digits are 5 and 9, their sum is 14 and their product is 45, and 14 + 45 = 59, so the output is 59 is a special two digit number.

Specimen run two. Input 42. The digits are 4 and 2, their sum is 6 and their product is 8, and 6 + 8 = 14, which is not 42, so the output is 42 is not a special two digit number.

Class header, main() and input of the number 2. Tens digit by division by 10 1. Units digit by remainder on 10 1. Condition comparing the sum plus the product with the number 2. Correct message in each case 1.

Question 9 [15]

(a) The completed class is as follows.

```
public class Area
{
    void compute(int s)
    {
        System.out.println("Area of the square = " + (s * s));
    }
    void compute(int l, int b)
    {
        System.out.println("Area of the rectangle = " + (l * b));
    }
    void compute(double b, double h)
    {
        System.out.println("Area of the triangle = " + (0.5 * b * h));
    }
    public static void main(String args[])
    {
        Area ob = new Area();
        ob.compute(6);
        ob.compute(8, 5);
        ob.compute(12.0, 5.0);
```

```
    }
}
```

Output: Area of the square = 36, then Area of the rectangle = 40, then Area of the triangle = 30.0, each on a line of its own. The third value carries a decimal point because the calculation is done in double arithmetic.

The compiler chooses the method whose parameter list matches the number and the types of the arguments in the call, so compute(6) goes to the one int version, compute(8, 5) to the two int version and compute(12.0, 5.0) to the two double version.

Note for the examiner: the three methods differ in the number or the type of their parameters, which is what makes the overloading legal. Methods differing only in their return type would not compile.

compute(int) with the correct area and output 3. compute(int, int) with the correct area and output 3. compute(double, double) with half the base times the height 3. main() creating one object and making all three calls 3. Three output lines, including the 30.0 2. Explanation of how the compiler picks the method 1.