

ISC Class 12 Computer Science 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: 70 | Time Allowed: Three hours | Reading Time: 15 minutes

Paper 1 (Theory) carries 70 marks. Part I is compulsory. In Part II attempt two questions from Section A, two from Section B and one from Section C.

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. Part I is compulsory and carries 20 marks. Part II carries 50 marks and is divided into three Sections. From Section A answer any two of the three questions, from Section B answer any two of the three questions, and from Section C answer any one of the two questions. Marks for each question, or part of a question, are shown in brackets. The programming language is Java, and all programs and program fragments must be written in Java. Give meaningful names to your variables, and indent your code so that the structure of every block is clear. No circuit diagrams, Karnaugh maps, truth tables, topology sketches or linked list figures are printed in this paper. Where a question asks for one you are expected to draw it yourself in pencil and label it. This paper carries 70 of the 100 marks for the subject, the remaining 30 being for the practical work of Paper 2.

PART I (20 Marks)

Answer all questions. Question 1 carries 10 marks and Question 2 carries 10 marks.

Question 1 [10]

Each part carries 1 mark. For the parts that offer four choices, write only the option you choose. For the rest, answer in a single sentence or give the single word, name, value or expression asked for.

- (i) The Boolean expression that is the complement of $A + B$, written without a bar over more than one letter, is:
(a) $A' + B'$ (b) $A'B'$
(c) AB (d) $A' + B$
- (ii) State the principle of duality as it applies to a Boolean identity.
- (iii) The decimal value of the hexadecimal number $(3A7)_{16}$ is:
(a) 827 (b) 935 (c) 967 (d) 1031
- (iv) Name the register of the central processing unit that holds the address of the instruction which is to be fetched next.
- (v) The gate whose output is 1 only when an odd number of its inputs are 1 is:
(a) NAND (b) NOR (c) XOR (d) AND
- (vi) Name the access modifier that allows a member of a class to be used inside that class and inside its subclasses, but not by an unrelated class in another package.
- (vii) A method declared in an interface, before any default method was permitted, is by its nature:
(a) private and static (b) public and abstract
(c) protected and final (d) private and final
- (viii) State the order in which elements leave a queue, and give the usual three letter name for that order.
- (ix) The number of usable host addresses available on a network whose mask is /26 is:
(a) 64 (b) 30 (c) 126 (d) 62
- (x) The right of the creator of a piece of original software to decide how it may be copied, changed and distributed is protected by:

- (a) copyright (b) a patent on the algorithm
 (c) a trademark on the file name (d) the licence key alone

Question 2 [10]

Each part carries 2 marks.

- (i) Convert the binary number $(1101101)_2$ into its decimal equivalent and into its hexadecimal equivalent. Show the working for both. [2]
- (ii) Find the complement of the Boolean expression $F = A + B'C$, and reduce the complement to a product of two factors. State the law you use at each step. [2]
- (iii) Give the output of the following two statements, and give a reason for the first value. [2]
`String s = "COMPUTER";`
`System.out.println(s.indexOf('T') + s.length());`
`System.out.println(s.substring(3).toLowerCase());`
- (iv) State two differences between an abstract class and an interface in Java. [2]
- (v) A stack of capacity five already holds the values 3, 8 and 12, with 12 at the top. Write down, in order, the values that are returned when two pop operations are carried out, and state the number of elements left in the stack. Then state what the condition `top equals minus one` tells the program. [2]

PART II (50 Marks)

Answer two questions from Section A, two questions from Section B and one question from Section C.

SECTION A

Answer any two questions from this Section. Each question carries 10 marks.

Question 3 [10]

- (a) A hostel gate releases a student only when a certain combination of three conditions is met. The three inputs are A, the warden's approval, B, a valid leave slip, and C, an emergency override. The output F is 1 when the gate opens. The gate opens whenever the warden approves and a leave slip is held, or whenever the warden does not approve but the emergency override is on, or whenever a leave slip is held and the override is on. Write the Boolean expression for F as a sum of three product terms, then reduce it to two product terms using Boolean laws alone, naming the law used at each step. [4]
- (b) Draw the logic circuit for the reduced expression you obtained in part (a) using only AND, OR and NOT gates, and label every gate and every line. State the total number of gates your circuit uses. [3]
- (c) Carry out the subtraction 53 minus 29 in eight bit binary using the 2's complement method. Show the eight bit form of each number, the 1's complement and the 2's complement of the number being subtracted, the addition and the fate of the carry that comes out of the leftmost column. Give the answer in decimal. [3]

Question 4 [10]

- (a) A function F of the four variables A, B, C and D is 1 for the input combinations numbered 1, 3, 5, 7, 8, 9, 10, 11 and 14, counting the combinations from 0 to 15 with A as the most significant variable. Reduce F to its simplest sum of products form using a Karnaugh map for four variables. Draw the map yourself, using Gray code ordering along both edges, mark every square that holds a 1, and ring each group you use, stating the size of each group and the term it yields. [5]
- (b) A second function G of the same four variables is 0 for the input combinations numbered 0, 1, 2, 3, 8, 10, 12 and 14, and is 1 for every other combination. Reduce G to its simplest product of sums form using a Karnaugh map of zeros. Draw the map, ring each group of zeros, and state the sum term each group yields. [5]

Question 5 [10]

- (a) Explain what a half adder does. Draw its truth table yourself, with columns for the two inputs A and B and for the two outputs, sum and carry, and fill in all four rows. Write the Boolean expression for each of the two outputs, and name the single gate that produces the sum output on its own. [4]
- (b) Explain why a half adder cannot be used on its own to add the middle columns of two multi bit binary numbers, and state what a full adder has that a half adder lacks. Write the Boolean expression for the carry output of a full adder as a sum of three product terms, and state how many half adders and how many further gates are needed to build one full adder. [3]
- (c) Describe the instruction cycle carried out by the central processing unit, naming each of its phases in order. For the fetch phase, name the two registers that are used and state the part each one plays. [3]

SECTION B

Answer any two questions from this Section. Each question carries 10 marks. In this Section write your answers in Java.

Question 6 [10]

- (a) Define a class named Library as described below, and write a main method that uses it. [10]

Class name: Library

Data members: title of type String to hold the name of the book, days of type int to hold the number of days for which the book has been kept, and fine of type double to hold the fine payable. All three are to be private.

Library(String t, int d): a parameterised constructor to set the title and the number of days, and to set the fine to zero

void computeFine(): to work out the fine on these rules. For seven days or fewer there is no fine. For the eighth day up to the twenty first day the fine is 2.0 rupees for every day beyond the seventh. Beyond the twenty first day the fine is 28.0 rupees for the first fourteen chargeable days together with 5.0 rupees for every day beyond the twenty first.

double getFine(): to return the fine

void display(): to print the title, the number of days and the fine, each on a line of its own, in that order

Write a main method that creates one object for a book titled Wings of Fire kept for 25 days and another for a book titled Panchatantra kept for 5 days, calls computeFine and display on each of them in turn, and finally prints the total of the two fines on a line of its own. State the complete output the program produces.

Question 7 [10]

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

- (a) Give the output of the following fragment, which is the body of main, and state the values held by a and b when the loop ends. [3]

```
int a = 2, b = 3, c = 0;
for (int i = 1; i <= 4; i++)
{
    c = a + b;
    a = b;
    b = c;
    System.out.print(c + " ");
}
```

```
System.out.println();
System.out.println(a + "," + b);
```

- (b)** Give the output of the following fragment, and explain in one sentence what the insert call does to the order of the characters. **[4]**

```
String w = "ALGORITHM";
StringBuilder sb = new StringBuilder();
for (int i = 0; i < w.length(); i++)
{
    char ch = w.charAt(i);
    if ("AEIOU".indexOf(ch) >= 0)
        sb.append(Character.toLowerCase(ch));
    else
        sb.insert(0, ch);
}
System.out.println(sb);
System.out.println(sb.length());
```

- (c)** Give the output of the following recursive method for the two calls shown, and state the base case of the recursion. **[3]**

```
static int f(int n)
{
    if (n <= 1)
        return 1;
    return n * f(n - 2);
}

public static void main(String args[])
{
    System.out.println(f(7));
    System.out.println(f(8));
}
```

Question 8 **[10]**

- (a)** Write a complete Java program that reads ten integers typed by the user into a single dimensional array and then, in a single pass over the array, finds the largest value, the second largest value and the average of the ten numbers. Print the three results, each on a line of its own, with the average printed as a real number. Do not sort the array and do not use any ready made library method for finding a maximum. State the output your program gives for the input 45, 12, 78, 3, 96, 23, 90, 7, 34 and 19. **[6]**
- (b)** Write a complete Java program that works through a four element array of dividends and a three element array of divisors, dividing the dividend at each position by the divisor at the same position and printing each quotient. Use a try block with one catch for an attempt to divide by zero and a separate catch for an index that lies outside an array, and use a finally block that prints a line at the end of every pass whether or not an exception was thrown. Take the dividends as 10, 20, 30 and 40 and the divisors as 2, 0 and 5, and state the complete output. **[4]**

SECTION C

Answer any one question from this Section. The question carries 10 marks.

Question 9 [10]

- (a) Define a class named `IntStack` that holds a stack of integers in an array, as described below. [7]

Data members: `st`, an array of `int` to hold the elements, `top`, an `int` to hold the index of the element at the top, and `cap`, an `int` to hold the capacity. All three are to be private.

`IntStack(int n)`: a parameterised constructor that creates the array with `n` places, stores the capacity and sets `top` so that the stack is empty

`void push(int v)`: to put `v` on the top of the stack if there is room, and otherwise to print a message that reports the overflow and names the value that could not be pushed

`int pop()`: to remove and return the element at the top of the stack, and, if the stack is already empty, to print a message reporting the underflow and to return minus 9999

Write a main method that creates a stack of capacity three, pushes 11, 22, 33 and then 44 in that order, and then prints the value returned by each of four successive `pop` calls. State the complete output.

- (b) A singly linked list of integers is held with a reference named `head` pointing at the first node. Each node holds a data field and a link field. Draw the structure of such a list holding the three values 5, 8 and 12 in that order, labelling `head`, both fields of every node and the value held in the link field of the last node. Then state the two statements, in order, that are needed to insert a new node pointed at by `n` immediately after the first node of the list, and explain in one sentence what goes wrong if those two statements are written in the opposite order. [3]

Question 10 [10]

- (a) Explain what is meant by cloud computing. Name and describe the three service models by which it is usually supplied, giving one example of the sort of thing a customer receives under each. Then state two risks that a school takes on when it moves its records to a cloud service. [5]
- (b) Distinguish between artificial intelligence and machine learning, and state the part that training data plays in the second. Explain what is meant by bias in a trained model, and state one duty that falls on the person who builds such a model. [3]
- (c) Explain what the Internet of Things is, and state one privacy concern peculiar to it. Name the Act in force in India that deals with offences committed with the help of computers and computer networks. [2]

ANSWER KEY AND MARKING SCHEME

Full code is given for every question that asked for a program or a class, and every program in this key has been compiled and run, so the outputs quoted are the outputs actually produced. 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. For a Boolean reduction accept any form that is logically equivalent to the one given and uses no more terms. For a drawing accept any clear and correctly labelled version.

Question 1 [10]

- (i) (b) $A'B'$. By De Morgan's first theorem the complement of a sum is the product of the complements, so $(A + B)' = A'B'$.
- (ii) The principle of duality states that a Boolean identity stays true if every OR is changed to an AND, every AND is changed to an OR, every 0 is changed to a 1 and every 1 is changed to a 0, the variables themselves being left as they are. The dual of $A + 1 = 1$ is therefore $A \cdot 0 = 0$.
- (iii) (b) 935. Working: $3 \times 256 + 10 \times 16 + 7 = 768 + 160 + 7 = 935$.
- (iv) The program counter, also called the instruction pointer. It holds the address of the next instruction to be fetched and is incremented once that instruction has been fetched.
- (v) (c) XOR. The exclusive OR gate gives 1 when the number of inputs at 1 is odd, which for two inputs means when the inputs differ.
- (vi) protected. A protected member is reachable inside its own class, inside any subclass wherever that subclass lives, and inside other classes of the same package, but not from an unrelated class in a different package.
- (vii) (b) public and abstract. Such a method carries no body, and both modifiers are supplied by the compiler whether or not they are written out.
- (viii) Elements leave a queue in the order in which they arrived, the element that has waited longest being removed first. The order is called FIFO, that is first in first out.
- (ix) (d) 62. A /26 mask leaves six bits for the host part, which gives 2 to the power 6, that is 64 addresses, and two of those are reserved, one for the network itself and one for the broadcast address, leaving 62 usable.
- (x) (a) copyright. Copyright arises in the source code and the object code as original literary work the moment it is written, and it governs copying, adaptation and distribution.

1 mark for each correct part. No mark is deducted for a wrong option, and a one-line answer that carries the correct idea gains the mark even if it is loosely worded.

Question 2 [10]

- (i) Decimal. Reading the bits from the left with their place values: $1 \times 64 + 1 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 64 + 32 + 8 + 4 + 1 = 109$. Hexadecimal. Group the bits in fours from the right, padding on the left: 0110 1101. The group 0110 is 6 and the group 1101 is 13, which is D. So $(1101101)_2 = (109)_{10} = (6D)_{16}$.

Decimal 109 with the place values shown 1. Hexadecimal 6D with the bits grouped in fours 1.

- (ii) $F = A + B'C$, so $F' = (A + B'C)'$. By De Morgan's first theorem the complement of a sum is the product of the complements, giving $F' = A' \cdot (B'C)'$. By De Morgan's second theorem the complement of a product is the sum of the complements, so $(B'C)' = B'' + C' = B + C'$, using the law of involution, that the complement of a complement is the variable itself. Hence $F' = A'(B + C')$, a product of the two factors A' and $B + C'$.

De Morgan's first theorem applied to the sum 1. De Morgan's second theorem with involution, giving $A'(B + C')$ 1.

- (iii) 13 on the first line and puter on the second line. The capital T in COMPUTER stands at index 5, counting from zero, and the length of the string is 8, so the first statement prints $5 + 8$, which is 13. Both operands are of type int, so the plus sign adds rather than joins. The call substring(3) takes every character from index 3 to the end, which is PUTER, and toLowerCase turns it into puter. The original string is not altered.

13 with the reason that indexOf gives 5 and length gives 8 1. puter 1.

- (iv) An abstract class may hold instance variables, constructors and fully written methods alongside its abstract ones, while an interface holds only abstract methods together with constants that are public, static and final. A

class may extend only one abstract class but may implement any number of interfaces, so an interface is the way a class takes on more than one set of obligations. A third acceptable difference is that an abstract class can declare members as private or protected, while every member of an interface is public by default.

One correct difference 1. A second correct difference 1.

- (v) The first pop returns 12 and the second pop returns 8, in that order, because a stack is last in first out and 12 was pushed most recently. One element, the value 3, is left in the stack. The condition top equals minus one tells the program that the stack is empty, so that a further pop must be refused as an underflow rather than attempted.

12 then 8, with one element left 1. top equals minus one recognised as the empty condition 1.

Question 3 [10]

- (a) The three conditions give one product term each. The warden approves and a leave slip is held gives AB. The warden does not approve but the override is on gives A'C. A leave slip is held and the override is on gives BC. So $F = AB + A'C + BC$. The reduction runs as follows. Expand the redundant term over A, since $A + A' = 1$ by the complement law, so $BC = BC(A + A') = ABC + A'BC$. This gives $F = AB + A'C + ABC + A'BC$. Now group the first with the third and the second with the fourth: $AB + ABC = AB(1 + C) = AB$, by the identity $1 + C = 1$ and then $AB.1 = AB$, and $A'C + A'BC = A'C(1 + B) = A'C$ in the same way. Hence $F = AB + A'C$. The term BC is redundant, which is the consensus theorem, and the laws used in order are the complement law, the distributive law, the identity $1 + X = 1$ and the law of identity $X.1 = X$.

Note for the examiner: the expression has been verified over all eight input combinations, and $AB + A'C + BC$ and $AB + A'C$ agree on every one of them.

$F = AB + A'C + BC$ written from the three conditions 2. Reduction to $AB + A'C$ 1. Laws named at the steps, or the consensus theorem named 1.

- (b) The circuit for $F = AB + A'C$ should be drawn as follows. Bring in the three input lines A, B and C at the left. Take A through a NOT gate, label the gate NOT 1 and label its output line A'. Feed A and B into a two input AND gate, label it AND 1 and label its output AB. Feed A' and C into a second two input AND gate, label it AND 2 and label its output A'C. Feed the two AND outputs into a two input OR gate, label it OR 1, and label its output F. The circuit uses four gates in all, one NOT, two AND and one OR.

NOT gate on A with the two AND gates correctly fed 2. OR gate combining the two products, with F labelled, and the total of four gates stated 1.

- (c) In eight bits, $53 = 00110101$ and $29 = 00011101$. The 1's complement of 29 is obtained by inverting every bit, giving 11100010. Adding 1 gives the 2's complement of 29, which is 11100011. Now add this to 53: $00110101 + 11100011 = 1\ 00011000$, that is a carry of 1 out of the leftmost column with 00011000 left in the eight bit result. In 2's complement subtraction a carry out of the leftmost column is simply discarded, and its presence shows that the answer is positive. The result 00011000 is $16 + 8 = 24$, so $53 \text{ minus } 29 = 24$, which is correct.

Note for the examiner: this arithmetic has been checked numerically. Accept a candidate who writes the addition column by column rather than in one line.

Eight bit forms of 53 and 29, with the 1's complement of 29 1. 2's complement 11100011 and the addition 1. Carry discarded with the answer 24 in decimal 1.

Question 4 [10]

- (a) List the combinations as binary values of ABCD. The function is 1 at 0001, 0011, 0101, 0111, 1000, 1001, 1010, 1011 and 1110. Draw a four variable Karnaugh map as a four by four grid, the columns headed by CD in the Gray code order 00, 01, 11, 10 and the rows headed by AB in the same order 00, 01, 11, 10, and place a 1 in each of those nine squares.

Three groups cover every 1. First group, a block of four made of the squares 0001, 0011, 0101 and 0111. In all four A is 0 and D is 1, while B and C each take both values, so the term is $A'D$. Second group, a block of four made of the squares 1000, 1001, 1010 and 1011. In all four A is 1 and B is 0, so the term is AB' . Third group, a pair made of the squares 1010 and 1110. In both A is 1, C is 1 and D is 0, while B changes, so the term is ACD' .

The simplified sum of products form is therefore $F = A'D + AB' + ACD'$.

Note for the examiner: this reduction has been checked by machine over all sixteen input combinations and agrees with the given function on every one. The square 1010 is deliberately covered twice, which is allowed and is what lets the pair be formed. Accept $A'D + AB' + ACD'$ in any order of terms.

Map drawn with Gray code ordering on both edges and the nine squares marked 2. Group of four giving $A'D$ 1. Group of four giving AB' 1. Pair giving ACD' 1.

- (b) For a product of sums form the zeros are grouped and each group yields a sum term in which every variable appears complemented if it is 1 throughout the group and uncomplemented if it is 0 throughout. The zeros lie at 0000, 0001, 0010, 0011, 1000, 1010, 1100 and 1110. Draw the same four by four map and place a 0 in each of those eight squares.

Two groups of four cover all eight zeros. First group, the squares 0000, 0001, 0011 and 0010, which form the whole $AB = 00$ row. In all four A is 0 and B is 0, so the sum term is $A + B$. Second group, the squares 1000, 1010, 1100 and 1110. In all four A is 1 and D is 0, while B and C each take both values, so the sum term is $A' + D$.

The simplified product of sums form is therefore $G = (A + B)(A' + D)$.

Note for the examiner: this reduction has been checked by machine over all sixteen combinations. The second group wraps round the map, the squares 1000 and 1010 lying in the $AB = 10$ row and 1100 and 1110 in the $AB = 11$ row, which are adjacent rows in Gray code ordering. Accept the two factors written in either order.

Map of zeros drawn and the eight squares marked 2. Group giving $A + B$ 1. Group giving $A' + D$ 1. Answer written as a product of the two sums 1.

Question 5 [10]

- (a) A half adder is the combinational circuit that adds two single binary digits and produces two outputs, a sum bit and a carry bit. It takes no carry in from a column to its right, which is why it serves only for the least significant column. The truth table should be drawn with four columns, A , B , Sum and Carry, and four rows. With $A = 0$ and $B = 0$ the sum is 0 and the carry is 0. With $A = 0$ and $B = 1$ the sum is 1 and the carry is 0. With $A = 1$ and $B = 0$ the sum is 1 and the carry is 0. With $A = 1$ and $B = 1$ the sum is 0 and the carry is 1, since binary 1 plus 1 is 10. The expressions are $\text{Sum} = A'B + AB'$ and $\text{Carry} = AB$. The single gate that produces the sum on its own is the XOR gate, since $A'B + AB'$ is exactly A exclusive or B .

Note for the examiner: the expressions have been checked against the arithmetic over all four input combinations.

Half adder defined as adding two bits to give a sum and a carry 1. Truth table with all four rows correct 1. Sum = $A'B + AB'$ and Carry = AB 1. XOR named for the sum 1.

- (b) In every column except the rightmost there are three bits to be added, the two digits of that column together with the carry that came out of the column to its right. A half adder has only two inputs, so it has nowhere to receive that incoming carry, and a chain of half adders would lose it. What a full adder has that a half adder lacks is a third input, the carry in. Its carry output is $C_{out} = AB + B.C_{in} + A.C_{in}$, which is 1 whenever any two or more of the three inputs are 1. A full adder can be built from two half adders together with one OR gate, the first half adder adding A and B , the second adding that sum to the carry in and the OR gate combining the two carries that the half adders produce.

Note for the examiner: the carry expression has been checked against the arithmetic over all eight input combinations. Accept C_{out} written as $AB + C_{in}(A + B)$, which is the same function.

Reason that the middle columns have a third bit, the incoming carry 1. Carry in named as the third input, with $C_{out} = AB + B.C_{in} + A.C_{in}$ 1. Two half adders and one OR gate 1.

- (c) The instruction cycle is the sequence the processor repeats for every instruction, and its phases in order are fetch, decode, execute and store, the last being the writing back of the result. In the fetch phase the processor copies the next instruction from memory into the processor itself. The two registers used are the program counter, which holds the address of that instruction and is placed on the address bus and then incremented so that it points at the one after, and the instruction register, which receives the instruction word when it arrives over the data bus and holds it while the control unit decodes it. A memory address register and a memory buffer register are also accepted as the pair named, provided their parts are described correctly.

Phases named in order 1. Fetch phase described 1. Program counter and instruction register, with the part each plays 1.

Question 6 [10]

(a) The completed class is as follows.

```
public class Library
{
    private String title;
    private int days;
    private double fine;
    public Library(String t, int d)
    {
        title = t;
        days = d;
        fine = 0.0;
    }
    public void computeFine()
    {
        if (days <= 7)
            fine = 0.0;
        else if (days <= 21)
            fine = (days - 7) * 2.0;
        else
            fine = 28.0 + (days - 21) * 5.0;
    }
    public double getFine()
    {
        return fine;
    }
    public void display()
    {
        System.out.println("Title : " + title);
        System.out.println("Days : " + days);
        System.out.println("Fine : " + fine);
    }
    public static void main(String args[])
    {
        Library a = new Library("Wings of Fire", 25);
        a.computeFine();
        a.display();
        Library b = new Library("Panchatantra", 5);
        b.computeFine();
        b.display();
        System.out.println("Total fine = " + (a.getFine() + b.getFine()));
    }
}
```

Working for the given data. For 25 days the third rule applies, so the fine is $28.0 + (25 - 21) \times 5.0 = 28.0 + 20.0 = 48.0$. For 5 days the first rule applies, so the fine is 0.0. The total is 48.0.

The output, as produced by the compiled program, is these seven lines in order: Title : Wings of Fire, then Days : 25, then Fine : 48.0, then Title : Panchatantra, then Days : 5, then Fine : 0.0, then Total fine = 48.0.

Note for the examiner: the fine must be printed as 48.0 and not 48, because the field is of type double. The boundary days 7 and 21 fall to the rule that charges nothing extra for them, so a candidate whose condition reads $\text{days} < 8$ in place of $\text{days} \leq 7$ is not penalised. Award full marks for $2 * (\text{days} - 7)$ written in place of $(\text{days} - 7) * 2.0$, since the result is the same double value once it is stored in fine.

Class header with the three private data members 2. Parameterised constructor setting all three fields 2. computeFine with all three branches of the rule correct 3. getFine and display, with the three lines printed in the right order 1. main creating both objects and calling both methods on each 1. Total printed and the seven output lines correct, with 48.0 and 0.0 carrying their decimal points 1.

Question 7 [10]

- (a) Dry run. Before the loop a is 2, b is 3 and c is 0. First pass: $c = 2 + 3 = 5$, then a becomes 3, then b becomes 5, and 5 is printed. Second pass: $c = 3 + 5 = 8$, then a becomes 5, then b becomes 8, and 8 is printed. Third pass: $c = 5 + 8 = 13$, then a becomes 8, then b becomes 13, and 13 is printed. Fourth pass: $c = 8 + 13 = 21$, then a becomes 13, then b becomes 21, and 21 is printed.

Output, as produced by the compiled program: the first line reads 5 8 13 21 followed by a trailing space, since print adds a space after every value, and the second line reads 13,21.

When the loop ends a holds 13 and b holds 21. The fragment generates the Fibonacci sequence.

Values traced through the four passes 1. First line 5 8 13 21 1. a = 13 and b = 21 on the second line as 13,21 1.

- (b) The string ALGORITHM has nine characters. Each vowel is appended at the end of the builder in small letters, and each consonant is inserted at position 0, that is pushed in at the front, so the consonants come out in the reverse of the order in which they were met while the vowels stay in their original order behind them. Taking the characters in turn: A is a vowel, so the builder becomes a. L is a consonant inserted at the front, giving La. G goes to the front, giving GLa. O is a vowel appended, giving GLao. R goes to the front, giving RGLao. I is appended, giving RGLaoi. T goes to the front, giving TRGLaoi. H goes to the front, giving HTRGLaoi. M goes to the front, giving MHTRGLaoi.

Output, as produced by the compiled program: MHTRGLaoi on the first line and 9 on the second line.

The insert call places the character at index 0 and shifts everything already in the builder one place to the right, so the consonants finish in reverse order.

Recognising that vowels are appended and consonants inserted at the front 1. Trace through the nine characters 1. MHTRGLaoi 1. Length 9 with the sentence explaining insert 1.

- (c) The method multiplies n by the result of the same method called on n minus 2, so it forms the product of every second number down from n. For $f(7)$: $7 \times f(5) = 7 \times 5 \times f(3) = 7 \times 5 \times 3 \times f(1)$, and $f(1)$ returns 1 because 1 is not greater than 1, so the value is $7 \times 5 \times 3 \times 1 = 105$. For $f(8)$: $8 \times f(6) = 8 \times 6 \times f(4) = 8 \times 6 \times 4 \times f(2) = 8 \times 6 \times 4 \times 2 \times f(0)$, and $f(0)$ returns 1 because 0 is not greater than 1, so the value is $8 \times 6 \times 4 \times 2 \times 1 = 384$.

Output, as produced by the compiled program: 105 on the first line and 384 on the second line.

The base case is n less than or equal to 1, where the method returns 1 without calling itself again. Note that the single test catches both chains, the odd one stopping at 1 and the even one stopping at 0.

105 with the chain shown 1. 384 with the chain shown 1. Base case stated as n less than or equal to 1 returning 1 1.

Question 8 [10]

- (a) The program is as follows.

```
import java.util.Scanner;
public class SecondLargest
{
    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();
    int big = a[0], second = a[0], total = a[0];
    for (int i = 1; i < 10; i++)
    {
        total = total + a[i];
        if (a[i] > big)
        {
            second = big;
            big = a[i];
        }
        else if (a[i] > second)
            second = a[i];
    }
    System.out.println("Largest = " + big);
    System.out.println("Second largest = " + second);
    System.out.println("Average = " + (double) total / 10);
}
}

```

Working for the given input 45, 12, 78, 3, 96, 23, 90, 7, 34, 19. The sum is $45 + 12 + 78 + 3 + 96 + 23 + 90 + 7 + 34 + 19 = 407$, so the average is 40.7. The largest is 96 and the value next below it is 90.

Output, as produced by the compiled program: Largest = 96, then Second largest = 90, then Average = 40.7, each on a line of its own.

Note for the examiner: the else if matters. If the second test were written as a separate if, a value equal to the new largest would wrongly be copied into second as well. The cast must be applied before the division, since $\text{total} / 10$ alone would be integer division and would print 40.0. Accept $\text{total} / 10.0$ in its place.

Array declared and the input loop reading ten values 1. Both trackers and the running total initialised from the first element 1. Single loop over the rest of the array 1. Largest updated with the old largest moved down into second 2. Second largest updated through an else if 1.

(b) The program is as follows.

```

public class ErrorTrap
{
    public static void main(String args[])
    {
        int a[] = {10, 20, 30, 40};
        int d[] = {2, 0, 5};
        for (int i = 0; i < 4; i++)
        {
            try
            {
                System.out.println(a[i] / d[i]);
            }
        }
    }
}

```

```

        catch (ArithmeticException e)
        {
            System.out.println("Cannot divide by zero");
        }
        catch (ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Index out of range : " + i);
        }
        finally
        {
            System.out.println("Pass " + i + " over");
        }
    }
}

```

Working. Pass 0 divides 10 by 2 and prints 5. Pass 1 divides 20 by 0, which throws an arithmetic exception.

Pass 2 divides 30 by 5 and prints 6. Pass 3 asks for d[3], which does not exist because d has only three places, so an array index exception is thrown. The finally block runs at the end of every pass.

Output, as produced by the compiled program, is these eight lines in order: 5, then Pass 0 over, then Cannot divide by zero, then Pass 1 over, then 6, then Pass 2 over, then Index out of range : 3, then Pass 3 over.

Note for the examiner: the two catch blocks may be written in either order here, since neither exception class is a parent of the other. The loop keeps running after an exception because the try block sits inside the loop rather than around it.

Both arrays declared and the loop running four times 1. try block holding the division with the two separate catch blocks 2. finally block printing on every pass 0.5. All eight output lines correct and in order 0.5.

Question 9 [10]

(a) The completed class is as follows.

```

public class IntStack
{
    private int st[];
    private int top;
    private int cap;
    public IntStack(int n)
    {
        cap = n;
        st = new int[cap];
        top = -1;
    }
    public void push(int v)
    {
        if (top == cap - 1)
            System.out.println("Overflow, cannot push " + v);
        else
        {
            top = top + 1;
            st[top] = v;
        }
    }
}

```

```

    }
}
public int pop()
{
    if (top == -1)
    {
        System.out.println("Underflow, the stack is empty");
        return -9999;
    }
    int v = st[top];
    top = top - 1;
    return v;
}
public static void main(String args[])
{
    IntStack s = new IntStack(3);
    s.push(11);
    s.push(22);
    s.push(33);
    s.push(44);
    System.out.println(s.pop());
    System.out.println(s.pop());
    System.out.println(s.pop());
    System.out.println(s.pop());
}
}

```

Working. The capacity is three, so 11, 22 and 33 are stored and top rises to 2. The push of 44 finds top equal to cap minus 1 and reports the overflow without storing anything. The four pops then return 33, 22 and 11 in that order, after which the stack is empty, so the fourth pop reports the underflow and returns minus 9999.

Output, as produced by the compiled program, is these six lines in order: Overflow, cannot push 44, then 33, then 22, then 11, then Underflow, the stack is empty, then -9999.

Note for the examiner: the underflow message is printed by pop itself and the value minus 9999 is printed by the println in main, which is why those two lines appear in that order. Accept a candidate who starts top at 0 and treats it as the count of elements, provided both the full and the empty tests are adjusted to match.

Three private data members with the constructor sizing the array and setting top to minus one 2. push testing for a full stack before storing, with top incremented first 2. pop testing for an empty stack, returning minus 9999 with a message 2. main with the four pushes and four pops, and all six output lines correct and in order 1.

- (b)** The drawing should show three boxes in a row, each divided into two parts, the left part holding the data and the right part holding the link. Write head to the left of the first box with an arrow running into it. The first box holds 5 in its data part and an arrow from its link part to the second box. The second box holds 8 and an arrow from its link part to the third box. The third box holds 12, and its link part holds null, which must be labelled as such, since null is what marks the end of the list.

To insert the new node pointed at by n immediately after the first node, the two statements in order are n.link = head.link; followed by head.link = n;. If they are written the other way round, head.link is overwritten with n before the address of the old second node has been saved anywhere, so the rest of the list becomes unreachable and is lost, and the statement n.link = head.link then merely makes the new node point at itself.

Three nodes drawn with both fields labelled, head shown and null in the last link 1. The two statements in the correct order 1. Explanation of what is lost when the order is reversed 1.

Question 10 [10]

- (a) Cloud computing is the supply of computing resources, that is processing power, storage, networks and ready made software, as a service over the internet, the resources being held in a provider's data centre rather than on the customer's own premises. The customer rents what is needed, can ask for more or less of it at short notice, and usually pays by use.

The three service models are these. Infrastructure as a service, under which the customer receives bare computing resources such as virtual machines, raw storage and virtual networks, and installs and maintains the operating system and everything above it. Platform as a service, under which the customer receives a ready made environment in which to run programs, the provider looking after the operating system, the database engine and the runtime, so that the customer supplies only the application and its data. Software as a service, under which the customer receives a finished application reached through a browser, such as a web based mail or office suite, and maintains nothing at all.

Two risks a school takes on are these. First, the records leave the school's own control and sit on hardware it does not own, possibly in another country, so a breach at the provider, or a demand made on the provider by some other authority, exposes data the school is answerable for. Second, the service depends wholly on the link to the provider, so a failure of the connection or of the provider itself leaves the school unable to reach its own records, and a provider who raises prices or closes down leaves the school tied in, since moving the data elsewhere may be difficult. Loss of control over where backups are kept, and difficulty in meeting a duty to delete a pupil's data on request, are also accepted.

Cloud computing defined as resources supplied as a service over a network 1. Three models named 1. One example of what the customer receives under each 1. First risk 1. Second risk 1.

- (b) Artificial intelligence is the wider field, covering any technique by which a machine is made to carry out a task that would ordinarily need human intelligence, which includes systems built from rules written out by hand as well as systems that learn. Machine learning is one approach within that field, in which the machine is not given the rules at all but derives them for itself from examples, so that its behaviour is fixed by the data it was shown rather than by instructions a programmer wrote.

Training data plays the part that the rules play in the older approach. The model is shown a large set of examples, adjusts its internal parameters until its outputs on those examples are close to the answers supplied with them, and is then used on examples it has never seen. Whatever pattern is present in the training data becomes the pattern the model applies.

Bias in a trained model means a systematic tendency to treat some group or some case less favourably, arising not from any intention but from the data. If the examples under represent a group, or carry the marks of past unfair decisions, the model reproduces that skew and may even sharpen it, while appearing neutral because it is a machine. One duty on the builder is to test the model's behaviour separately on each group it will be applied to, rather than resting on an overall accuracy figure, and to document what the model was trained on and where it should not be relied on. Keeping a human decision maker in the loop where the outcome matters to a person is also accepted.

Artificial intelligence distinguished as the wider field, with machine learning as learning from examples 1. Part played by training data 1. Bias explained with one duty on the builder 1.

- (c) The Internet of Things is the arrangement by which ordinary objects, rather than computers, are fitted with sensors and a network connection, so that they collect readings and exchange them, and often act on them, without a person operating them. Lights, meters, watches, vehicles and medical monitors are the usual examples. The privacy concern peculiar to it is that these devices sit inside the places people treat as private, and collect continuously and invisibly, so a householder has no moment at which consent is asked and often no way to see what is being sent or to whom. Readings that seem harmless on their own, such as electricity use or a door sensor, together reveal when a house is empty and what its occupants do hour by hour. The Act in force in India is the Information Technology Act.

Internet of Things defined 1. One privacy concern peculiar to it, with the Information Technology Act named 1.