

CISCE SPECIMEN PAPER 2025-2026

COMPUTER APPLICATIONS

SOLUTION

Prepared by SPONDON GANGULI

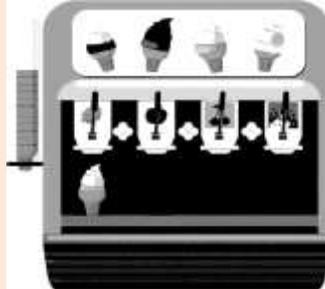
Question 1. MCQs

- (i) Which of the following *String methods* results into a *boolean* data type?
- (a) trim()
 - (b) equals()
 - (c) replace()
 - (d) concat()

Ans. Option (b) equals

Reason: equals() method returns a boolean data type

- (ii) Which *construct* can be used to get only one of the required ice creams?



- (a) switch construct
- (b) while construct
- (c) do.. while construct
- (d) for construct

Ans. Option (a) switch construct

Reason: The switch construct works on menu-driven or for selecting one out of many options.

- (iii) Which statement is *correct* for the *method prototype* given below:

int check(char ch, String s)

- (a) check() does not return any value
- (b) check() has return type int
- (c) check method has two actual parameters
- (d) check() is a constructor

Ans. Option (b) check() has return type int

Reason: check() method has two parameters, but they are formal parameters and not actual parameters, whereas the return type is int.

- (iv) The *Math method* which returns *int* value is:

- (a) round()
- (b) cbrt()
- (c) ceil ()
- (d) random()

Ans. Option (a) round()

Reason: Except the round() method, all other methods return a double value.

(v) What is the output of the following statement:

```
"MONOPOLY".lastIndexOf('O');
```

- (a) 1
- (b) 3
- (c) 2
- (d) 5

Ans. Option (d) 5

Reason: Last index of the letter 'O' is 5

(vi) Typecasting is often necessary when dealing with financial data. Identify the correct syntax to *typecast a double to an int* using the variables:

```
int amount;
```

```
double valueINR
```

- (a) `int amount = valueINR;`
- (b) `int amount = Integer.parseInt (valueINR);`
- (c) `int amount = (int) valueINR;`
- (d) `int amount = int (valueINR);`

Ans. Option (c) `int amount = (int) valueINR;`

Reason: Explicit typecasting needed here to convert a higher data type (double) to a lower data type (int).

(vii) What is the *output* of the code snippet given below?

```
int lives = 5;
```

```
System.out.print(lives--);
```

```
System.out.print(lives);
```

- (a) 4 3
- (b) 5 4
- (c) 5 3
- (d) 4 4

Ans. Option (b) 5 4

Reason: Post-decrement operator used, decrement took place after the print statement.

(viii) What will be the *output* of the following statement? String

```
s = "JavaProgramming";
```

```
System.out.println(s.substring(4, 11).toUpperCase());
```

- (a) Programm
- (b) PROGRAMM
- (c) PROGRAM
- (d) Program

Ans. Option (c) PROGRAM

Reason: `s.substring(4, 11)` will extract characters from index position 4 to 10 and `toUpperCase()` will convert that to uppercase.

- (ix) Which of the following *access specifiers* will make a member accessible *only* within its own class?
- (a) public
 - (b) private
 - (c) protected
 - (d) default

Ans. Option (b) private

Reason: private members can only be accessed within the class.

- (x) What will be the *output* of the following Java method?

Character.isLetterOrDigit('\n')

- (a) 1
- (b) 0
- (c) true
- (d) false

Ans. Option (b) false

Reason: '\n' represents new line. It does not belong to letters (A-Z) nor it belong to digits (0-9). So, the method will return false.

- (xi) Which of the following is *not* a type of *token* in Java?

1. Method	3. Literal
2. Identifier	4. Keyword

- (a) only 1
- (b) 1 and 3
- (c) only 2
- (d) only 4

Ans. Option (a) only 1

Reason: These are the 5 tokens in Java – Keywords, Identifiers, Literals, Operators and Punctuators.

- (xii) The statement given below is termed as:

public void Accept(int a)

- (a) Method signature
- (b) Method block
- (c) Method prototype
- (d) Constructor

Ans. Option (c) Method prototype

Reason: A method prototype represents the function name along with its return type and parameter list, and not the body, whereas a method signature is the number, type and sequence of parameters mentioned in the parameter list of the method prototype.

(xiii) What is the **output** of the following Java code?

```
boolean flag = false;  
if (flag) { System.out.println("True"); }  
else { System.out.println("False"); }
```

- (a) True
- (b) False
- (c) No output
- (d) Compilation error

Ans. Option (b) False

Reason: if(flag) represents whether if(flag==true). If true, it will return true; else return false. Here, flag=false, hence the else block will get executed.

(xiv) Identify the **static method** from the list given below:

- (a) length()
- (b) nextLine()
- (c) substring(int)
- (d) isLetter(char)

Ans. Option (d) isLetter(char)

Reason: All other methods need an object to call, like String_object.length (); String_object.substring(), Scanner_object.nextLine() but, but isLetter() is called directly by Character.isLetter(), where Character is a wrapper class. Hence, it must be a static method and not an instance method.

(xv) **String a[]={ "Rohini", "Rajarshi", "Rajeev", "Rehan", "Rebecca" };
System.out.println(a[2].substring(2));**

Which one of the following will be the output of the above statements?

- (a) jeev
- (b) Ra
- (c) Raj
- (d) Je

Ans. Option (a) jeev

Reason: a[2] represents 3rd element in the array and substring(2) will extract characters from the index 2 to the end.

(xvi) **System.out.println(Math.round(Math.ceil(-8.8)));** will result in:

- (a) 8.0
- (b) -8.0
- (c) -9
- (d) -8

Ans. Option (d) -8

Reason: Math.ceil() rounds up a decimal value to nearest whole number. So, -8.8 = -8.0 and Math.round() converts the value to an integer, thus -8.

(xvii) Which one of the following Java statements *assigns 100* to the *last* element of a 3×3 array?

- (a) `x[2][2]=100;`
- (b) `x[3][3]=100;`
- (c) `x[2][3]=100;`
- (d) `x[3][2]=100;`

Ans. Option (a) `x[2][2]=100;`

Reason: A 3×3 array will have index positions starting from `[0][0]` to `[2][2]`, which means the last row and last column index is 2,2.

(xviii)

 Floppy Disk	 Pen Drive	 DVD
 Hard Disk	 Storage Tape	 Memory card

Consider the *Two-dimensional array* $S[2][3]$, of *storage devices* given above, state the *index* of the *Hard disk*.

- (a) `S[1][0]`
- (b) `S[0][1]`
- (c) `S[1][2]`
- (d) `S[0][0]`

Ans. Option (a) `S[1][0]`

Reason: Hard disk is stored in the second row and first column; thus, the index position will be 1,0.

(xix) How many times is the *for* statement given below executed?

for ($k = 10; k \geq 0; k--$)

- (a) 10
- (b) 11
- (c) 12
- (d) 0

Ans. Option (b)

Reason: If the loop runs from 10 to 0, it counts a total of 11 iterations, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0.

- (xx) Consider the following program segment in which the statements are *jumbled*. Choose the *correct order* of the statements to return the *sum of the first 10 natural numbers*.

```
for(i=1; i<=10; i++)    1
return sum;             2
int sum = 0, i;         3
sum+=i;                 4
```

- (a) 1 2 3 4
(b) 3 4 1 2
(c) 1 4 2 3
(d) 3 1 4 2

Ans. Option (d) 3 1 4 2

Question 2.

- (i) Write *Java expression* for the following: $x^{10} + y^{10}$

Ans. `z = Math.pow(x,10) + Math.pow(y,10);`

- (ii) *Evaluate* the given expression when $x = 4$

$x^* = - - x + x - - + x;$

Ans. $x = x^* (-x + x - - + x);$
 $= 4*(3 + 3 + 2)$
 $= 4*8 = 32$

- (iii) Convert the following *switch* case into *if else if*:

```
switch(x)
{
case 'T' :
case 't' : System.out.print("Teacher");
break; default :
System.out.print("Student");
}
```

Ans.

```
if(x=='T' || x=='t')
    System.out.println("Teacher");
else
    System.out.println("Student");
```

- (iv) Write the *output* of the following program segment:

```
for( int a=1; a<=10;a++)  
{ if(a%2==0)  
    continue;  
System.out.print(a+" ");  
}
```

Output:1 3 5 7 9

Working –

```
a=1 print 1  
a=2 continue  
a=3 print 3  
a=4 continue  
a=5 print 5  
a=6 continue  
a=7 print 7  
a=8 continue  
a=9 print 9  
a=10 continue
```

- (v) In the example given below of class *Cat*, identify the *variable* and the *methods*:

```
Cat  
Name  
meow()  
eat()  
play()
```

Ans. class – Cat

variables – Name

methods – meow(), eat(), play()

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

```
int k = 100;
while (k >= 10)
{
    System.out.println(k); k -= 40;
}
```

Ans. loop runs 3 times

Output: 100

60

20

Working –

k=100 k>10 (true) print 100 k=k-40 = 60

k=60 k>10 (true) print 60 k=k-40=20

k=20 k>10 (true) print 20 k=k-40=-20

k=-20 k>10 (false)

Loop ends

- (vii) Consider the given array and answer the questions given below:

```
int z[][] = {{2, 3, 4}, {5, 1, 2}, {7, 9, 3}};
```

- (a) What is the *order* of the array *z[][]*?

Ans. 2D array

- (b) What is the value in *z[2][0]*?

Ans. 7

- (viii) Give the *output* of the following:

- (a) *“ROSE”.compareTo(“ROJA”)*

Ans. 9 (Difference : ASCII(‘S’) – ASCII(‘J’))

- (b) *“DEDICATE”.replace(‘D’, ‘L’)*

Ans. LELICATE

- (ix) Consider the following array and answer the questions given below:

```
char ch [] = { ‘A’, ‘%’, ‘y’, ‘@’, ‘7’, ‘p’};
```

- (a) How many *bytes* does the array occupy?

Ans. 12 bytes (1 char occupies 2 bytes, 2x6=12)

- (b) What is the output of the statement *Character.isDigit(ch[4])*?

Ans. true (ch[4] = ‘7’)

```
(x) class perform
{
    int m ;
    String name;

    perform( int x , String y) //constructor
    {
        m=x;
        name=y;
    }

    void print ()
    {
        System.out.print( name+ “ ” +m);
    }

    public static void main()
    {
        perform ob1=new perform( 95 , “Xavier”);
        ob1.print();
    }
}
```

(a) Give the *output* of the code given above.

Ans. Xavier 95

(b) Name the *type* of the *constructor* used.

Ans. parameterized constructor

Question 3

Define a class with following specifications.

class name: **Hotel**

Member variables:

String name: stores name of customer name

long mobno: stores mobile number

int days: stores number of days customer stayed in hotel

int bill: stores customer bill

Member method:

void input () – input values using Scanner class methods only

void charge() – calculate bill as per the following criteria

<u>days</u>	<u>charge/day</u>
first 3 days	1000 Rs/ day
next 4 days	900 Rs/day
> 7 days	800 Rs/day
bill = bill + gst (18% of bill)	

void print() - Display customer name, mobile number and bill.

Invoke all the above methods in the **main method** with the help of an object.

Solution:

```
import java.util.Scanner;
```

```
class Hotel
```

```
{
```

```
    String name;
```

```
    long mobno;
```

```
    int days;
```

```
    double bill;
```

```
    void input() // Accept customer details
```

```
    {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter the name of the customer");
```

```
        name = sc.nextLine();
```

```
        System.out.println("Enter the mobile no. of the customer");
```

```
        mobno = sc.nextLong();
```

```
        System.out.println("Enter the number of days stayed in the hotel");
```

```
        days = sc.nextInt();
```

```
    }
```

```

void charge() // Calculate bill with slab system and GST
{
    if(days <= 3)
        bill = days * 1000;
    else if(days <= 7)
        bill = (3*1000) + (days-3)*900;
    else
        bill = (3*1000) + (4*900) + (days-7)*800;
    bill += bill * 0.18;
}

void print() // Display details
{
    System.out.println(name);
    System.out.println(mobno);
    System.out.println(bill);
}

public static void main()
{
    Hotel h = new Hotel(); //creating an object of the class Hotel
    h.input(); //calling input() method using the object h
    h.charge(); //calling charge() method
    h.print(); //calling print() method
}
}

```

Variable Type Description

name	String	Customer name
mobno	long	Mobile number
days	int	Days stayed
bill	double	Total bill

Question 4

Define a *class* to accept values into a 3x3 integer array and print the *product of each row* elements. Example:

3	1	2
4	2	1
5	1	2

Output:

Row 0 – 6

Row 1 – 8

Row 2 – 10

Solution:

```
import java.util.Scanner;
class RowProduct
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        int a[][] = new int[3][3];
        // Input array elements
        for(int i=0;i<3;i++)
            for(int j=0;j<3;j++)
                a[i][j] = sc.nextInt();
        // Calculate row products
        for(int i=0;i<3;i++)
        {
            int prod = 1;
            for(int j=0;j<3;j++)
                prod *= a[i][j];
            System.out.println("Row "+i+" – "+prod);
        }
    }
}
```

Variable Type Description

a	int[][]	3×3 array
prod	int	Row product

Question 5

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

int transform(int n) – to return the sum of the digits of the given number

Example: n = 458 output : 17

void transform(String s) – to convert the given String to upper case and print

Example: if S = “Blue” Output : BLUE

void transform (char ch) – to print the character ch in **3 rows** and **3 columns** using nested loops.

Example: if ch = ‘@’ Output : @@@
 @@@ @@@

Solution:

```
class Transform
{
    int transform(int n)
    {
        int sum = 0;
        while(n > 0)
        {
            sum += n % 10;
            n /= 10;
        }
        return sum;
    }
    void transform(String s)
    {
        System.out.println(s.toUpperCase());
    }
    void transform(char ch)
    {
        for(int i=1;i<=3;i++)
        {
            for(int j=1;j<=3;j++)
                System.out.print(ch);
            System.out.println();
        }
    }
    public static void main()
    {
        Transform t = new Transform();
        System.out.println(t.transform(458));
        t.transform("Blue");
        t.transform('@');
    }
}
```

Variable	Type	Description
n	int	Number
s	String	Text
ch	char	Character

Question 6

Define a *class* to accept a string. Check if it is a *Special String* or not.

A String is Special if the *number of vowels equals to the number of consonants*.

Example: MINE

Number of vowels = 2

Number of Consonants = 2

Solution:

```
import java.util.Scanner;
class SpecialString
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine().toUpperCase();
        int v = 0, c = 0;
        for(int i=0;i<s.length();i++)
        {
            char ch = s.charAt(i);
            if(ch>='A' && ch<='Z')
            {
                if("AEIOU".indexOf(ch)>=0)
                    v++;
                else
                    c++;
            }
        }
        if(v == c)
            System.out.println("Special String");
        else
            System.out.println("Not a Special String");
    }
}
```

Variable	Type	Description
s	String	Input string
v	int	Vowel count
c	int	Consonant count

Question 7

Define a *class* to accept a number and check if the sum of the *first digit* and the *last digit* is an *even number* or an *odd number*. Example: N = 2396 N = 9316

First digit:	2	9
Last digit:	6	6
Sum:	8	15
Output:	Sum is even	Sum is odd

Solution:

```
import java.util.Scanner;
class DigitSumCheck
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int ld = n % 10;
        int fd = n;
        while(fd >= 10)
            fd /= 10;
        int sum = fd + ld;
        if(sum % 2 == 0)
            System.out.println("Sum is even");
        else
            System.out.println("Sum is odd");
    }
}
```

Variable	Type	Description
n	int	Number
fd	int	First digit
ld	int	Last digit

Question 8

Define a *class* to accept 10 integers in an array, search for the given value using the *Linear Search technique* and print appropriate messages.

Solution:

```
import java.util.Scanner;
class LinearSearch
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        int a[] = new int[10];
        boolean flag = false;
        for(int i=0;i<10;i++)
            a[i] = sc.nextInt();
        int key = sc.nextInt();
        for(int i=0;i<10;i++)
        {
            if(a[i] == key)
            {
                System.out.println("Element found at index "+i);
                flag = true;
                break;
            }
        }
        if(!flag)
            System.out.println("Element not found");
    }
}
```

Variable	Type	Description
a	int[]	Array
key	int	Search element
flag	boolean	Found status

~ THANK YOU ~