ICSE EXAMINATION PAPER - 2025
COMPUTER APPLICATIONS
Class-10th
(Solved)

Maximum Marks: 100 | *Time Allotted: Two Hours*

Instructions to Candidates:

1. *Answers to this Paper must be written on the paper provided separately.*
2. *You will **not** be allowed to write during first **15** minutes.*
3. *This time is to be spent in reading the question paper.*
4. ***The time given at the head of this Paper is the time allowed for writing the answers.***
5. *This Paper is divided into two Sections.*
6. *Attempt **all** questions from **Section A** and **any four** questions from **Section B.***
7. *The intended marks for questions or parts of questions are given in brackets[].*

SECTION A (40 Marks)

*(Attempt **all** questions from this **Section**.)*

Question 1 **[20]**

Choose the correct answers to the questions from the given options.
(Do not copy the questions, write only the correct answers.)

(i) Character class methods are found in the ***package*** called:

(a) java.util **(b)** java.lang
(c) java.awt **(d)** java.io

(ii) System.out.println(**'Z' + 32**); will display:

(a) z **(b)** Z
(c) 122 **(d)** 154

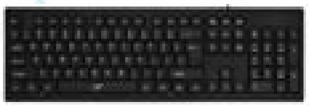
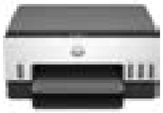
(iii) ***double x []–{2.5,4.5,5.5,6.4);*** occupies __________ bytes.

(a) 16 **(b)** 4
(c) 8 **(d)** 32

(iv) The output of ***42/6%2*** is:

(a) 1 **(b)** 10
(c) 2 **(d)** 0

(v)

 Mouse	 Keyboard	 Printer
 Scanner	 Barcode Scanner	 Microphone

Consider the ***Two dimensional array P/2/13],*** of peripherals (input / output devices) given above, state the index of the device ***Barcode Scanner.***

(a) P[1][1] **(b)** P[0][1]
(c) P[1][2] **(d)** P[0][0]

(vi) Which of the following is ***user defined data type***?

1. array	3. class
2. double	4. boolean

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

(vii) Select the ***infinite*** loop:

(a) for(int i=1;i<=10;i++) **(b)** for(int i=2; i!=0;i-=3)
(c) for(int i=5;i<=5;i++) **(d)** for(int i=1;i>=1;1−−)

(viii) The output of ***Math.max(−7, Math.min(−5,−4))*** is

(a) −5 **(b)** −4
(c) −7 **(d)** error

(ix) Which of the following is ***true*** for the given object creation statement?
Game cricket = new Game();

(a) Game is an object of cricket class **(b)** New keyword creates object Game
(c) Game is a class and cricket is an object **(d)** Game and cricket are objects

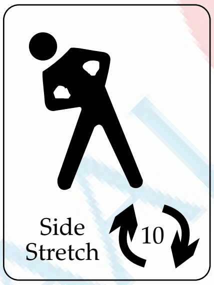
(x) *Post office* is an example for __________ access specifier.

(a) public **(b)** local
(c) protected **(d)** private

(xi) **Assertion (A):** In switch case ***break*** statement avoids fall through.
Reason (R): ***break*** statement helps to execute only one case at a time.

(a) Both (A) and (R) are true and (R) is a correct explanation of (A).
(b) Both (A) and (R) are true and (R) is not a correct explanation of (A).
(c) (A) is true and (R) is false.
(d) (A) is false and (R) is true.

(xii) A physical education teacher asks the students to do the side stretch is steer below, 10 times. Which ***programming construct*** the teacher uses.



(a) if **(b)** switch
(c) for **(d)** if else if

(xiii) The ***index (subscript)*** of the last element of an array ***ar[]*** is

(a) ar.length() **(b)** ar[].length
(c) ar.length()-1 **(d)** ar.length-1

(xiv) **Assertion (A):** A clock is a real-life example of ***nested loops.***
Reason (R): The hour hand moves through 12 positions, while the minute hand moves through 60 positions within each hour.

(a) Both (A) and (R) are true and (R) is a correct explanation of (A).
(b) Both (A) and (R) are true and (R) is not a correct explanation of (A).
(c) (A) is true and (R) is false.
(d) (A) is false and (R) is true.

(xv) Which of the following ***pairs of methods*** will cause a ***compile-time error*** due to incorrect method ***overloading***?

(a) void test(int a, int b) and void test(double a, double b)
(b) void test(int a, double b) and void test(double a, int b)
(c) void test(int a, double b) and void test(int a)
(d) void test(int a) and int test(int a)

(xvi) Which of the following converts *“25”* to *25.0*?

(a) Double. Parsedouble(“25”)

(b) Double.parse(“25”)

(c) Double.parseDouble(“25”)

(d) Double.parseDouble(25)

(xvii) Consider the program segment:

```
int p=0;
for(p=4; p>0; p−=2);
System.out.print(p);
System.out.println(p);
```

The above statements will display:

(a) 42

(b) 4200

(c) 0
0

(d) 00

(xviii) *System.out.println(“I said, “It’s wise to obey elders.\””);*

The *output* of the above statement is:

(a) I said,’lt is wise to obey elders.’

(b) I said, “It’s wise to obey elders.”

(c) I said, It’s wise to elders.

(d) “‘It’s wise to obey elders.’”

(xix) What is the *output* of the statement given below?

“ANGER”.compareTo(“ANGEL”)

(a) 3

(b) −6

(c) 6

(d) 0

(xx) Consider the following program segment in which the statements are *jumbled.* Choose the *correct order* of statements to calculate and return the *factorial of 4.*

```
for (k=1; k<=4; k++)      →  1
return fa;                →  2
long fa 1, k;             →  3
fa*=k;                    →  4
```

(a) 1, 2, 3, 4

(b) 3, 1, 4, 2

(c) 3, 1, 2, 4

(d) 1, 3, 2, 4

Question 2

(i) Write the *java expression* to find the product of square root of **P** and the square root of **Q** using the methods of *Math class.* **[2]**

(ii) Write the *output* of the following String method: **[2]**

```
String x= “talent”; String y=”matrix”;
System.out.print(x.substring(3).concat(y.substring(3));
```

(iii) Write the Java statement for creating an object named *‘sifra’* of the class *‘Robot’,* which takes *three double parameters.* **[2]**

(iv) Convert the given loop into **exit controlled loop.** **[2]**

```
int a,b;
for (a=10, b=1; a>=1; a−2)
{
b+= a;
b++;
}
System.out.print(b);
```

(v) Consider and give the *output* of the following program: **[2]**

```
class report
{ int a,b;
   report()
   { a=10;
     b=15;
{
report(int x, int y)
{ a=x;
  b=y;
}
void print()
{ System.out.println(a*b);
}
static void main()
{ report r=new report();
    r.print);
    report p = new report(4, 5);
    p.print();
    }
}
```

(vi) **(a)** Name one ***String method*** which results in ***positive integer only.*** **[2]**

(b) Name one ***String method*** which results in a ***character.***

(vii) John was asked to write a Java code to calculate the ***surface area of a cone,*** the following code was written by him: **[2]**

Surface area of cone is A=πrl $l=\sqrt{r^2+h^2}$

```
class area
{ double area (double r, double h)
    { double 1, a;
      a=22.0/7*r*l;
      1=Math.sqrt(r*r+h* h);
      return a;
    }
}
```

Specify the ***type of the error*** in the above program, correct and write the program to be error free.

(viii) Consider the following array and answer the questions given below: **[2]**

int a[] = {12, 10, 8, 4, 6 , 2, 3, 5, 7}

(a) What is the ***output*** of System.out.print*(a[0]+a[5]);*?

(b) What is the ***index (subscript)*** of the ***largest*** element of the array ***a[]***?

(ix) **(a)** Write the Java statement to initialise the ***first 6 odd numbers*** in a 3 × 2 array. **[2]**

(b) What is the result of ***x[0][1]+x[2][1]*** of the above array?

(x) Give the ***output*** of the following program segment and specify ***how many times*** the loop is executed. **[2]**

```
String s = "JAYA";
for(i=0;i<s.length();i+=2)
System.out.println(s.substring(i);
```

SECTION B (60 Marks)

(*Attempt* ***any four*** *questions from this* **Section**.)

The answers in this section should consist of the programs in either BlueJ environment or any program environment with Java as the base.

Each program should be written using variable description / mnemonic codes so that the logic of the program is clearly depicted.

Flowcharts and algorithms are not required.

Question 3 **[15]**

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

- **Member Variables:**
  - int acno – stores the user's account number.
  - int space – stores the amount of storage space in GB purchased by the user.
  - double bill – stores the total price to be paid by the user.
- **Member Methods:**
  - void accept() – prompts the user to input their account number and storage space using Scanner class methods only.
  - void calculate() – calculates the bill total price based on the storage space purchased using the pricing table provided:

Storage range	Price per GB (Rs)
First 15 GB	15
Next 15 GB	13
Above 30 GB	11

  - void display() – displays the account number, storage space and bill to be paid.

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

Question 4 **[15]**

Define a *class* to accept values into a 4 × 4 integer array. Calculate and print the ***NORM*** of the array.

NORM is the square root of sum of squares of all elements.

1	2	1	3
5	2	1	6
3	6	1	2
3	4	6	3

Sum of squares of elements = 1 + 4 + 1 + 9 + 25 + 4 + 1 + 36 + 9 + 36 + 1 + 4 + 9 + 16 + 36 + 9 = 201

NORM Squareroot of 201 = 14.177446878757825

Question 5 **[15]**

Define a *class* to accept a String and Print if it is a ***Super string*** or not. A ***String is Super*** if the ***number of uppercase letters*** are equal to the ***number of lower case letters.***

[Use Character & String methods only]

Example: "COmmlTmeNt"

Number of Uppercase letters –5

Number of Lowercase letters –5

String is a Super String

Question 6 **[15]**

Define a *class* to initialise the following data in an array.

Search for a given character input by the user, using the ***Binary Search technique.***

Print "***Search Successful***" if the character is found otherwise print "***Search is not Successful***".

'A', 'H', 'N', 'P', 'S', 'U', 'W', 'Y', 'Z', 'b', 'd'

Question 7 **[15]**

Define a *class* to overload the method ***print()*** as follows:

void print()– To print the given format using nested loops.

```
@#@#@
@#@#@
@#@#@
@#@#@
@#@#@
```

double print(double *a*, double *b*)– To display the sum of numbers between ***a*** and ***b*** with difference of 0.5.

e.g. if *a*=1.0, *b*=4.0

output is: 1.0 + 1.5 + 2.0 + 2.5 + ... +4.0

int print(char ch1, char ch2)– compare the two characters and return the ASCII code of the largest character.

Question 8 **[15]**

Define a *class* to accept a number. Check if the sum of the ***largest digit*** and the ***smallest digit*** is an even number or an odd number. ***Print appropriate messages.***

Sample Input:	6425	3748
Largest digit:	6	8
Smallest digit:	2	3
Sample Output:	Sum is even	Sum is odd

Answers

SECTION A

1. **(i)** Option (b) is correct.

Explanation: In Java, the Character class methods, which provide functionalities for manipulating characters, are located in the java.lang package.

(ii) Option (c) is correct.

Explanation: 'Z' + 32 means ASCII code is 90.

90 + 32 = 122, which is printed.

(iii) Option (d) is correct.

Explanation: The array carries [2, 5, 4.5, 5.5, 6.4], which means 2 integers and 3 doubles, So, total memory occupied will be 2 × 4 + 3 × 8 = 8 + 24 = 32 bytes.

(iv) Option (a) is correct.

Explanation: 42/6%2 = 7%2 = 1.

(v) Option (a) is correct.

Explanation: The element is in row index 1 and column index 1, so P[1, 1].

(vi) Option (b) is correct.

Explanation: Arrays and classes are user defined data types in java. Both can hold multiple data elements.

(vii) Option (b) is correct.

Explanation: The loop is for (i=2; I!=0; I−=3), starting from value 2 , if I is decreased by 3 every time, it will never reach 0, hence it is an infinite loop.

(viii) Option (a) is correct.

Explanation: Math.min(−5, −4) gives −5 and Math.max(−7, −5) gives −5.

(ix) Option (c) is correct.

Explanation: Cricket is an object of the Game class, created using the new operator in the statement Game cricket = new Game();

(x) Option (a) is correct.

Explanation: A **public** access specifier allows a class, method, or variable to be accessed from anywhere in the program.

A **Post Office** is a public service that is accessible to everyone, similar to how the **public** access modifier makes members accessible to all parts of a program.

(xi) Option (a) is correct.

Explanation: Because of the break statement control of execution goes out of the switch….case construct and fall through is avoided and also the valid matching case(s) are executed.

(xii) Option (c) is correct.

Explanation: Since the work done here is repetitive in nature, the for programming construct is used here.

(xiii) Option (d) is correct.

Explanation: ar.length gives the last position and −1 gives the actual index, as the index of the array starts from 0 to n−1.

(xiv) Option (a) is correct.

Explanation: Since the hour hand moves 12 times and with each iteration the minute hand moves for 60 times, it is a nested loop, loop inside loop.

(xv) Option (d) is correct.

Explanation: Java does not consider the return type for method overloading. By changing the return type does not mean that the method is overloaded. Hence, void test (int a) and int test (int a) causes a compile time error.

(xvi) Option (c) is correct.

Explanation: The Double.parseDouble("25") converts the string "25" to double 25.0.

(xvii) Option (d) is correct.

Explanation: Since there are no statements in the loop, no values of p are printed inside the loop. When the loop ends p has become 0.

The last two print statements print the value of p, that is 00.

(xviii) Option (b) is correct.

Explanation: The '\' escape sequence hides the special meaning of character.

(xix) Option (c) is correct.

Explanation: Here the compareTo() compares the strings "ANGEL" and "ANGER" and gives the difference in ASCII codes of the differentiating characters 'L' and 'R':

"ANGER" vs "ANGEL":

- 'A' == 'A' → **equal**
- 'N' == 'N' → **equal**
- 'G' == 'G' → **equal**
- 'E' == 'E' → **equal**
- 'R' vs 'L' → **'R' (82) − 'L' (76) = 6**

(xx) Option (b) is correct.

Explanation: First the variables are initialised. Next the loop starts, and then the value of k is repeatedly multiplied with the return variable 'fa'. Finally, the return value is returned.

2. **(i)** Math.sqrt(P) * Math.sqrt(Q)

(ii) 'entrix"

Explanation: x.substring(3) returns characters from index 3 to the end of x, giving "ent", similarly y.substring(3) returns characters from index 3 to the end of y giving 'rix'. Both are concatenated.

(iii) Robot Sifra=new Robot(77.213, 182.668, 919.817);

(iv)
```
int a,b;
a=10;
b=1;
do
{
  b+=a;
  b++;
  a-=2;
} while(a>=1);
System.out.print(b);
```

(v) 150

20

Explanation: When the 1st time report class object r is created, the default constructor is called and a, b are initialised with 10,15. Thus the print() function prints 150.

2nd time when the object p is created, the parameterised constructor is called to initialize a,b with values 4,5. Thus the print() function gives 20.

(vi) (a) length(): Returns the number of characters in a string.

(b) charAt(): Returns the character at the index specified.

(vii) Logical Error: Area has been calculated before calculation of l.

```
class area
{ double l, a;
  double area(double r, double h)
  {
    l=Math.sqrt(r*r + h*h)
    a=22.0/7*r*l;
    return a;
  }
}
```

(viii) (a) 14.

Explanation: a[0] is 12, a[5] is 2.

(b) 0.

Explanation: Largest element is 12 at index 0.

(ix) (a) int ar[][]={{1,3},{5,7},{9,11}};

(b) 14.

Explanation: x[0][1] = 3, x[2][1]=11

3 + 11 = 14.

(x) JAVA

VA

Loop executes for 2 times.

Explanation: The string "JAVA" has length 4 and the loop variable increases by 2 every time, hence loop goes for 2 times.

Substring i shows strings from index i to the end of the string. (i takes the values 0 and 2 hence the substrings shown are "JAVA" and "VA".

SECTION B

3.

```
import  java.util.scanner;
class CloudStorage
{
  int acno;
  int space;
  double bill;
public void accept():
{
  Scanner sc=new Scanner(System.in);
  System.out.println("Enter your account number :");
  acno=sc.nextInt();
  System.out.println("Enter amount of storage space required :");
  space=sc.nextInt();
sc.close();
}
public void calculate()
{
  if (space<=15)
    bill=space*15;
  else if(space<=30)
    bill=15*15 + ((space-15)*13));
  else if (space>30)
    bill=15*15 + 15 *13 + ((space-30)*11);
}
public void display()
{
  System.out.println("Account No.: " + acno);
  System.out.println("Storage       :" + space);
  System.out.println("Bill : "+ bill);
}
public static void main(String[]  args)
{
  CloudStorage ob=new CloudStorage();
  ob.accept();
  ob.calculate();
  ob.display();
  }
}
```

4.

```
import java.util.Scanner;
public class Main {
public static void main(String[] args) {
  double sum=0.0;
  double norm=0.0;
  Scanner scanner = new Scanner(System.in);
  int[][] matrix = new int[4][4];
  System.out.println("Enter numbers for 4x4 matrix: ");
    for (int i = 0; i < 4; i++) {
      for (int j = 0; j < 4; j++) {
        matrix[i][j] = scanner.nextInt();
          }
}
for (int i = 0; i < 4; i++) {
  for (int j = 0; j < 4; j++) {
    sum+=matrix[i][j] * matrix[i][j];
}
}
 norm=Math.sqrt(sum);
  System.out.println("Norm of the matrix is :"+
   norm);
   Scanner.colse();
}
}
```

5.
```
import java.util.*;
public class SuperString {
   public static void main(String[] args) {
      int uppercaseCount = 0, lowercaseCount = 0;
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter string: ");
      String sen = sc.nextLine();
      for (int i = 0; i < sen.length(); i++) {
         char ch = sen.charAt(i);
         if (Character.isLowerCase(ch)) {
            lowercaseCount++;
         } else if (Character.isUpperCase(ch)) {
            uppercaseCount++;
   }
}
System.out.println("Number of Uppercase letters: "
+ uppercaseCount);
System.out.println("Number of Lowercase letters: "
+ lowercaseCount);
if (uppercaseCount == lowercaseCount) {
   System.out.println("String is a Super String");
} else {
   System.out.println("String is NOT a Super String");
}
sc.close();
}
}
```

6.
```
import java.util.*;
public class BinarySearch
{
int binarySearch(char a[], int l, int r, char x)
   {
      while (l <= r) {
         int m = (l + r) / 2;
         if (a[m] == x) {
            return m;
         } else if (a[m] > x) {
            r = m - 1;
         } else {
            l = m + 1;
         }
      }
      return -1;
}
public static void main(String args[])
{
   BinarySearch ob = new BinarySearch();
   Scanner sc=new Scanner(System.in);
   char arr[]= {'A','H','N','P','S','U','W','Y','Z','b','d'};
   int n = arr.length;
   System.out.println("Enter search character :");
   char c=sc.nextLine().charAt(0);
   int res = ob.binarySearch(arr, 0, n - 1, c);
   if (res == -1)
      System.out.println("Search is not successful");
   else
      System.out.println("Search successful , Element
      found at index " + res);
   Sc.close();
   }
}
```

7.
```
import java.util.*;
public class Overload {
// Method to print the given pattern
public void print() {
   for (int i = 0; i < 4; i++) {
      for (int j = 0; j < 4; j++) {
         if (j % 2 == 0)
            System.out.print("@");
         else
            System.out.print("#");
   }
   System.out.println(); // New line after each row
   }
}
// Method to display the sum of numbers between a
and b with step 0.5
public void print(double a, double b) {
   double sum = 0.0;
   System.out.print("Output: ");
   for (double x = a; x <= b; x += 0.5) {
      System.out.print(x + " + ");
      sum += x;
   }
   System.out.println("\b\b = " + sum); // Removes
   last `+ ` and prints the sum
}
// Method to return ASCII code of the larger
character
public int print(char ch1, char ch2) {
   return (ch1 > ch2) ? (int) ch1 : (int) ch2;
}
public static void main(String args[]) {
   Overload obj = new Overload();
   // Testing pattern printing
   obj.print();
   // Testing sum function
   obj.print(1.0, 4.0);
   // Testing character comparison
   System.out.println("Larger character ASCII: " +
   obj.print('A', 'Z'));
   }
}
```

8.
```
import java.util.*;
public class LargestSmallest
{
public static void main(String args[])  {
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter number :");
    int d,lg=0,sg=0,num;
    num=sc.nextInt();
    lg=num%10;
    sg=num%10;
    while (num>0)
    {
        d=num%10;
        if (d>lg)
            lg=d;
        if (d<sg)
            sg=d;
        num/=10;
    }
    System.out.println("Largest digit :"+lg);
    System.out.println("Smallest digit :"+sg);
    if ((sg+lg)%2==0){
        System.out.println("Sum of largest and smallest digits is even");
    }else {
        System.out.println("Sum of largest and smallest digits is odd");
        }
    }
    }
```