

Java Programming

Data Structures & Algorithm



Class No : 1

Solve Puzzles :

Q1. A chocolate costs ₹1 each.

For every 3 wrappers, you can get 1 additional chocolate for free.

If you have ₹40, find out how many total chocolates you can eat, using the above scheme optimally.

Q2. You hire a worker for 7 days and agree to pay them 1 unit of gold at the end of each day. However, you only have a single 7-unit gold bar. The conditions are: The worker must be paid exactly 1 unit per day. You are allowed to cut the gold bar into pieces. After cutting, you can use those pieces to make exact payments each day. What is the minimum number of cuts required to split the gold bar so that you can pay the worker correctly for all 7 days?

Q3. Given two hourglasses of 4 minutes and 7 minutes, the task is to measure 9 minutes.

Q4. Four people — A, B, C, and D — need to cross a bridge at night.

They have only one torch, and the bridge is too dangerous to cross without it.

At most two people can cross at a time, and when two people cross together, they must move at the slower person's speed.

The time each person takes to cross the bridge is as follows:

A → 1 minute






B → 2 minutes

C → 7 minutes

D → 13 minutes

Class No : 2

Flow Chart

| Symbol | Name | Function |
|---|--------------|--|
|  | Start/end | An oval represents a start or end point |
|  | Arrows | A line is a connector that shows relationships between the representative shapes |
|  | Input/Output | A parallelogram represents input or output |
|  | Process | A rectangle represents a process |
|  | Decision | A diamond indicates a decision |

Q1. Add Two Numbers

Q2. Simple Interest Calculation

Q3. Check Whether a Number is Odd or Even

Q4. Find the Maximum of Two Numbers

Q5. Find the Maximum of Three Numbers

Q6. Grade Card Program (Based on Marks)

Q7. Print Counting from 1 to N

Q8. Find the Sum of N Natural Numbers

Q9. Print the Multiplication Table of 7

Q10. Find the Factorial of a Number

Q11. Print All Even Numbers from 1 to 100 using a Loop

Q12. Find the Sum of Digits of a Given Number using a Loop

Q13. Reverse a Number using a While Loop

Class No : 3

Installation of Java (JDK) & eclipse

JDK = JVM + Compiler + Libraries + Tools

IDE? Integrated Development Environment

IDE vs Editor

Can we use VS Code

Creating a new project in Eclipse

Writing your first program –

main- public static void main(String[] args) {}

sout- System.out.println("Welcome to coding Blocks");

Running a Java program

Comments

Single Line //

Multi line /* comments */

Understanding the need of a variable

Variables

can contain Data or Object References

Variable declaration, Initialization

Naming Convention for **Class/Variable/Method name** - identifiers

Must start with an alphabet or _ or \$ or not with numbers

Can end with an alphabet or _ or \$ or numeric digit

Spaces are not allowed

No reserved keyword

Intro to data types

Data Types :

Primitive or Pre-defined

Primitive :

Integer :

byte a = 127; // 1 byte, range: -128 to 127

short b = 32000; // 2 bytes, range: -32,768 to 32,767

int c = 100000; // 4 bytes, range: -2 billion to 2 billion

long d = 100000000000L; // 8 bytes, very large numbers (add L at end)

Decimal :

float e = 3.14f; // 4 bytes, 6-7 decimal digits (add f at end)

double f = 3.14159; // 8 bytes, 15 decimal digits

Others

char g = 'H'; // 2 bytes, single character

boolean h = true; // 1 bit, true or false

Non-Primitive :

User defined or Custom data types

String literals

String name = "Hansraj"

String - stores combination of characters Ex - "Coding Blocks"

Arrays,

Game of brackets

() – Methods - Parentheses

{ } - Scope/body - Curly

[] - Array - Square

Operators :

Arithmetic +, -, *, /, %, ++, --

++/-- operators can only be applied on variables, not on constants

Special powers of / & % by powers of 10

/ to reduce the number

% to get last digit(s) of number

Relational , <= , >= , == , !=

Assignment = , += , -= , *= , /= , % =

Logical &&, || , !

Used to combine multiple conditions

Rest to be covered later like bitwise and shift operators

Questions :

Code for All Flowcharts in Lecture 2

Q1. Lottry game

You are given a number n .

You have to determine what the person wins based on the following conditions:

If $n \geq 300$ and $n \leq 460$, the prize is MacBook.

If $n \geq 200$ and $n \leq 280$, the prize is Kurkure.

If $n \geq 1100$ and $n \leq 1500$, the prize is Cycle.

If $n > 50$ and $n \leq 80$, the prize is Bike.

For all other values of n , print "Better luck next time."

Q2. Lottery Game – Part 2

You are given a number n .

Based on the value of n , print what the person wins using the following conditions:

MacBook Range

If $n \geq 300$ and $n \leq 460$,

Print "You won a MacBook!"

Then check:

If $n \geq 300$ and $n \leq 380$, print "Model: M1 Mac"

If $n \geq 381$ and $n \leq 460$, print "Model: M2 Mac"

Kurkure Range

If $n \geq 200$ and $n \leq 280$,

Print "You won a pack of Kurkure!"

Then check:

If $n \geq 200$ and $n \leq 240$, print "Flavor: Chilli Kurkure"

If $n \geq 241$ and $n \leq 280$, print "Flavor: Onion Kurkure"

Cycle Range

If $n \geq 1100$ and $n \leq 1500$,
Print "You won a Cycle!"

Then check:

If $n \geq 1100$ and $n \leq 1300$, print "Brand: Avon Cycle"

If $n \geq 1301$ and $n \leq 1500$, print "Brand: Hero Cycle"

Bike Range

If $n > 50$ and $n \leq 80$,
Print "You won a Bike!"

Then check:

If $n > 50$ and $n \leq 65$, print "Model: Bullet"

If $n > 65$ and $n \leq 80$, print "Model: Rajdoot"

Otherwise

For all other values of n ,
Print "Better luck next time."

Class No : 4,5,6,7,8

Pattern

Star Pattern

Ques 1:

n = 5

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Ques 2:

n = 5

```
*
* *
* * *
* * * *
* * * * *
```

Ques 3:

n = 5

```
* * * * *
* * * *
* * *
* *
*
```


Ques 4:

n = 5

```

                *
            *   *
        *   *   *
    *   *   *   *
*   *   *   *   *
```

Ques 5:

n = 5

```

*   *   *   *   *
    *   *   *   *
        *   *   *
            *   *
                *   *
                    *
```

Ques 6:

n = 5

```

*   *   *   *   *
        *   *   *   *
            *   *   *
                *   *   *
                    *   *
                        *   *
                            *
                                *
```

Ques 7:

n = 5

```
*   *   *   *   *
*           *
*           *
*           *
*   *   *   *   *
```

Ques 8:

n = 5

```
*           *
    *       *
      *
    *       *
*           *
```

Ques 9:

n = 5

```
           *
        *   *   *
      *   *   *   *
    *   *   *   *   *
  *   *   *   *   *   *
*   *   *   *   *   *   *
```

Ques 10:

n = 5

```
*  *  *  *  *  *  *  *  *
  *  *  *  *  *  *  *
    *  *  *  *  *
      *  *  *
        *
```

Ques 11:

n = 5

```
      *
    *  *
  *    *
*      *
*      *
*      *
```

Ques 12:

n = 5

```
      *
    *  !  *
  *    !  *
*      !  *
*      !  *
*      !  *
*      !  *
```

Ques 13:

n = 5

```
*
*  *
*  *  *
*  *  *  *
*  *  *  *  *
*  *  *  *
*  *  *
*  *
*
```

Ques 14:

n = 5

```


      *
    *  *
  *  *  *
*  *  *  *
  *  *  *  *
    *  *  *
      *  *
        *  *
```

Ques 15:

n = 5

```
*   *   *   *   *
      *   *   *   *
            *   *   *
                  *   *
                        *
                              *
                                  *
                                      *
                                          *
                                              *
                                                  *
                                                      *
                                                          *
                                                              *
                                                                  *
                                                                      *
```

Ques 16:

n = 5

```
*   *   *   *   *
      *   *   *   *
            *   *   *
                  *   *
                        *
                              *
                                  *
                                      *
                                          *
                                              *
                                                  *
                                                      *
                                                          *
                                                              *
                                                                  *
                                                                      *
```

Ques 17:

n = 7

```

*   *   *       *   *   *
*   *           *   *
*               *
*
*           *
*   *       *   *
*   *   *   *   *   *

```

Ques 18:

n = 7

```

          *
        * * *
      * * * * *
    * * * * * * *
      * * * * *
        * * *
          *

```

Ques 19:

n = 7

```

*   *   *   *   *   *   *
*   *   *       *   *   *
*   *           *   *
*               *
*           *   *   *
*   *       *   *   *
*   *   *   *   *   *
*   *   *   *   *   *

```

Ques 20:

n = 7

```
      *
     *
    *
   *
  *
 *
*
```

Ques 21:

n = 5

```
*
* *
* * *
* * * *
* * * * *
```

Ques 22:

n = 5

```
* * * * *
* * * *
* * *
* *
*
```

Number Pattern

Ques 23:

n = 5

```

        1
      1 1 1
    1 1 1 1 1
  1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
```

Ques 24:

n = 5

```

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

Ques 25:

n = 5

```

        1
      2 3 4
    5 6 7 8 9
  10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25
```


Ques 26:

n = 5

```

        1
      1 2 3
    1 2 3 4 5
  1 2 3 4 5 6 7
1 2 3 4 5 6 7 8 9
```

Ques 27:

n = 5

```

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

Ques 28:

n = 5

```

        1
      2 3 2
    3 4 5 4 3
  4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

Ques 29:

n = 5

```

        1
      2 0 2
    3 0 0 0 3
  4 0 0 0 0 0 4
5 0 0 0 0 0 0 0 5
```

Ques 30:

n = 5

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

Ques 31:

n = 5

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

Ques 32:

n = 5

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

Ques 33:

n = 10

```
0
9 0 9
8 9 0 9 8
7 8 9 0 9 8 7
6 7 8 9 0 9 8 7 6
5 6 7 8 9 0 9 8 7 6 5
4 5 6 7 8 9 0 9 8 7 6 5 4
3 4 5 6 7 8 9 0 9 8 7 6 5 4 3
2 3 4 5 6 7 8 9 0 9 8 7 6 5 4 3 2
1 2 3 4 5 6 7 8 9 0 9 8 7 6 5 4 3 2 1
```