Subject: Professional Development Skills	Lecture - 5	Date: 27.12.2024 / Friday	No. of Programs : 04

Consider the currency system in which there are notes of six denominations namely Re 1, Rs 2, Rs 5, Rs 10, Rs 50, Rs 100. If a sum of Rs N is entered as input. Write a c , java, python program to compute the smallest number of notes that will combine to give Rs N.

C

```c
#include <stdio.h>

void min_notes(int n) {
    int d100 = n / 100;
    n %= 100;

    int d50 = n / 50;
    n %= 50;

    int d10 = n / 10;
    n %= 10;

    int d5 = n / 5;
    n %= 5;

    int d2 = n / 2;
    n %= 2;

    int d1 = n;

    printf("Minimum number of notes required: %d\n", d100 + d50 + d10 + d5 + d2 + d1);
}

int main() {
    int n;
    printf("Enter the amount in Rs: ");
    scanf("%d", &n);
    min_notes(n);
    return 0;
}
```

JAVA

```java
import java.util.Scanner;

public class MinNotes {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the amount in Rs: ");
        int n = sc.nextInt();

        int d100 = n / 100;
        n %= 100;

        int d50 = n / 50;
        n %= 50;

        int d10 = n / 10;
        n %= 10;

        int d5 = n / 5;
        n %= 5;

        int d2 = n / 2;
        n %= 2;

        int d1 = n;

        System.out.println("Minimum number of notes required: " + (d100 + d50 + d10 + d5 + d2 + d1));
    }
}
```

PYTHON

```python
def min_notes(n):
    d100 = n // 100
    n %= 100

    d50 = n // 50
    n %= 50

    d10 = n // 10
    n %= 10

    d5 = n // 5
    n %= 5

    d2 = n // 2
    n %= 2

    d1 = n // 1

    return d100 + d50 + d10 + d5 + d2 + d1

# Input and Output
n = int(input("Enter the amount in Rs: "))
print("Minimum number of notes required:", min_notes(n))
```

Time Complexity:

Each denomination is handled in a constant number of steps using arithmetic operations. **Time Complexity: O(1).**

Space Complexity:

Only a fixed number of variables are used to store the counts and remainders. **Space Complexity: O(1).**

If a five digit number is input , write a c , java, python program to print a new number by adding one to each of its digits

```
12345
11111
_________
23456
```

C

```c
#include <stdio.h>

void add_one_to_digits(int num) {
    int result = 0, multiplier = 1;

    while (num > 0) {
        int digit = num % 10;
        int new_digit = (digit + 1) % 10; // Add 1 and wrap
        result = result + (new_digit * multiplier);
        multiplier *= 10;
        num /= 10;
    }

    printf("New number: %d\n", result);
}

int main() {
    int num;
    printf("Enter a five-digit number: ");
    scanf("%d", &num);
    add_one_to_digits(num);
    return 0;
}
```

JAVA

```java
import java.util.Scanner;

public class AddOneToDigits {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a five-digit number: ");
        int num = sc.nextInt();

        String result = "";
        while (num > 0) {
            int digit = num % 10;
            int newDigit = (digit + 1) % 10; // Add 1 and wrap
            result = newDigit + result;
// Build the new number as a string
            num /= 10;
        }

        System.out.println("New number: " + result);
    }
}
```

PYTHON

```python
def add_one_to_digits(num):
    # Convert the number to a string to access digits
    result = ""
    for digit in str(num):
        new_digit = (int(digit) + 1) % 10
# Add 1 and handle digit wrapping
        result += str(new_digit)
    return int(result)

# Input and Output
num = int(input("Enter a five-digit number: "))
print("New number:", add_one_to_digits(num))
```

Time Complexity:
Each solution processes all 5 digits of the input number sequentially.
Time Complexity: O(d), where ddd is the number of digits (5 in this case).
For a fixed 5-digit input, this simplifies to **O(1).**

Space Complexity:
Python: The space used depends on the length of the string representation of the number, which is constant for 5 digits.
Space Complexity: O(1).
Java and C: Use only integer variables and a few constants for processing.
Space Complexity: O(1).

If cost price and selling price of an item is input, write a program to determine whether the seller has made profit or incurred loss. Also determine how much profit he made or loss he incurred.

C

```c
#include <stdio.h>

int main() {
    float costPrice, sellingPrice, result;

    printf("Enter Cost Price: ");
    scanf("%f", &costPrice);
    printf("Enter Selling Price: ");
    scanf("%f", &sellingPrice);

    if (sellingPrice > costPrice) {
```

JAVA

```java
import java.util.Scanner;

public class ProfitOrLoss {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter Cost Price: ");
        float costPrice = sc.nextFloat();
        System.out.print("Enter Selling Price: ");
        float sellingPrice = sc.nextFloat();
```

PYTHON

```python
def calculate_profit_or_loss(cost_price, selling_price):
    if selling_price > cost_price:
        print(f"Profit: {selling_price - cost_price:.2f}")
    elif selling_price < cost_price:
        print(f"Loss: {cost_price - selling_price:.2f}")
    else:
        print("No Profit No Loss.")

# Input
cost_price = float(input("Enter Cost Price: "))
selling_price = float(input("Enter Selling Price: "))
```

result = sellingPrice - costPrice;
        printf("Profit: %.2f\n", result);
    } else if (sellingPrice < costPrice) {
        result = costPrice - sellingPrice;
        printf("Loss: %.2f\n", result);
    } else {
        printf("No Profit No Loss.\n");
    }
 	
    return 0;
}
	if (sellingPrice > costPrice) {
	            System.out.printf("Profit: %.2f\n", sellingPrice - costPrice);
	        } else if (sellingPrice < costPrice) {
	            System.out.printf("Loss: %.2f\n", costPrice - sellingPrice);
	        } else {
	            System.out.println("No Profit No Loss.");
	        }
	
	        sc.close();
	    }
	}
		calculate_profit_or_loss(cost_price, selling_price)

Time Complexity

- **Input Analysis**: Determining profit/loss requires comparing sellingPrice and costPrice.
- **Operations**: At most one subtraction and comparison.
- **Conclusion: O(1)** for all three programs.

Space Complexity

- **Variables Used**:
  - costPrice and sellingPrice (2 variables).
  - **Additional Memory**: None (no data structures or dynamic memory allocation).
- **Conclusion: O(1)** for all three programs

Any integer is input, write a c, java and python program to find out whether it is even or odd number.

C	JAVA	PYTHON

#include <stdio.h>
 	
int main() {
    int number;
 	
    printf("Enter an integer: ");
    scanf("%d", &number);
 	
    if (number % 2 == 0) {
        printf("%d is an even number.\n", number);
    } else {
        printf("%d is an odd number.\n", number);
    }
 	
    return 0;
}
	import java.util.Scanner;
	
	public class EvenOrOdd {
	    public static void main(String[] args) {
	        Scanner sc = new Scanner(System.in);
	
	        System.out.print("Enter an integer: ");
	        int number = sc.nextInt();
	
	        if (number % 2 == 0) {
	            System.out.println(number + " is an even number.");
	        } else {
	            System.out.println(number + " is an odd number.");
	        }
	        sc.close(); } }
		def check_even_or_odd(number):
		    if number % 2 == 0:
		        print(f"{number} is an even number.")
		    else:
		        print(f"{number} is an odd number.")
		
		# Input
		number = int(input("Enter an integer: "))
		check_even_or_odd(number)

Time Complexity

Operation: The % (modulus) operator is used to check if the number is divisible by 2. **Conclusion: O(1)** for all three programs.

Space Complexity

Variables Used: Only one variable number is used to store the input integer, **Additional Memory**: None (no data structures or dynamic memory allocation). **Conclusion: O(1)** for all three programs.