

L2. Conditionals and Loops + Calculator Program

Conditional Statements

1. If-else Statement

if else statements are used to check a condition and execute a block of code if the condition is true or else they must execute a default block of code if the condition is false.

Syntax:

```
if (boolean expression T or F){  
    //body  
}  
else{  
    //body  
}
```

Example Program:

```
System.out.print("Enter your salary: ");  
int salary = sc.nextInt();  
System.out.println("Your salary currently is: " + salary);  
if(salary >= 2000){  
    salary = salary + 2000;  
    System.out.println("You recieved a 2000 INR bonus!");  
}  
else{  
    salary = salary + 1000;  
    System.out.println("You recieved a 1000 INR bonus!");  
}  
  
System.out.println("Your monthly paycheck: " + salary);
```

Note: You can obviously also have an if statement without an else statement **BUT** not an else statement without a if statement.

2. if-else if statements

Else if statements are statements which are used to check more than one condition

Syntax:

```
if (boolean expression T or F){  
  //body  
}  
else if(boolean expression T or F){  
  //body  
}  
else{  
  //body  
}
```

Example Program:

```
System.out.print("Enter your salary: ");  
int salary = sc.nextInt();  
System.out.println("Your salary currently is: " + salary);  
if(salary >= 10000){  
    salary += 2000;  
    System.out.println("You recieved a 2000 INR bonus!");  
}  
else if(salary >= 20000){  
    salary += 4000;  
}  
else{  
    salary += 1000;  
    System.out.println("You recieved a 1000 INR bonus!");  
}  
  
System.out.println("Your monthly paycheck: " + salary);
```

Note: Only the first condition is executed since even if the salary is greater than or equal to 20000 it still passes the first condition and only that condition is executed.

We also have nested if-else statements and so on, that is just and if else in the block of code of another if-else code if the if condition is true that block of code containing if-else statement is executed.

Loop Statements

Loops are used to execute a block of code multiple times until the condition is false.

1. For Loop

Syntax:

```
for (initialization; condition; increment/decrement){  
//Body;  
}
```

Example Program:

```
Scanner sc = new Scanner(System.in);  
System.out.println("Enter a number: ");  
int n = sc.nextInt();  
  
for (int i = 1; i <= n; i++){  
    System.out.println(i);  
}
```

How it works is while "i" (which has the value of 0) is less than N it increments it by 1 until it meets the condition. which is when i is atleast greater than N.
N is any number.

2. while loops

Syntax:

```
while (condition){  
//body  
}
```

Example Program:

```
int num = 1;  
while(num <= 5){  
    System.out.println(num);  
    num++;  
}
```

Note: The reason why we have while loops when we already have a for loop is because we use a for loop if we know how many times the loop will iterate, we use a while loop when we don't know how many times the loop will iterate.

3. do while loop

The do while loop is a special type of loop that executes the block of code atleast once before checking the condition of the loop.

Syntax:

```
do{  
//body  
}  
while (Condition);
```

Example Program:

```
int num = 1;  
do{  
System.out.println("It iterates atleast ONCE!");  
}  
while(num != 1);
```

Questions

1. Three integers are given to you find the largest of the three numbers

```
int a = sc.nextInt();  
int b = sc.nextInt();  
int c = sc.nextInt();  
  
int max = a;  
if (b > max){  
    max = b;  
}  
else if(c > max){  
    max = c;
```

OR

Using the math library

```
int max = Math.max(c, Math.max(a, b, c));
```

2. Write a program to check if the character is a lower case or upper case character.

```
char ch = sc.next().trim().charAt(0);  
//String str = sc.nextLine();  
if (ch >= 'a' && ch <= 'z'){  
    System.out.println("lowercase");  
}  
else{
```

```
        System.out.println("uppercase");  
    }  
}
```

3. Write a program to give the nth Fibonacci number.

```
int N = sc.nextInt();  
int count = 2;  
int first = 0;  
int second = 1;  
  
while(count <= N){  
    int temp = second;  
    second = first + second;  
    first = temp;  
    count++;}  
System.out.println();  
System.out.println("The nth number is " + second);
```

The above program gives us the nth Fibonacci number using the temp_case logic

We can also get the sequence with this code:

```
int N = sc.nextInt();  
int count = 2;  
int first = 0;  
int second = 1;  
  
System.out.print(first + " " + second);  
  
while(count <= N){  
    int temp = second;  
    second = first + second;  
    first = temp;  
    System.out.print(" " + second);  
    count++;}  
System.out.println();  
System.out.println("The nth number is " + second);
```

BOOM SHAKALAKA BOOM YOU GOT IT!!!

4. Write a program to count the number of occurrences of a single digit in a number.

```

System.out.print("Enter a long number ig: ");
int N = sc.nextInt();
int Unit = 0;
int count = 0;
while(N > 0) {
    Unit = N % 10;
    N /= 10;
    if (7 == Unit) {
        count++;
    }
}
System.out.println("The count of the number in the bigger nummber is: "
+ count);

```

The above program uses the number modulo 10 trick to find the unit digit and then checks if it is equal to 7.

because we wanna find how many 7's are there in the number.

5. Write a program to reverse a given number N.

```

System.out.println("Enter a number: ");
int N = sc.nextInt();

int result = 0;
while(N > 0){
    int unit = N%10;
    N /= 10;
    result += unit;
    if(N != 0) {
        result *= 10;
    }
}
System.out.println("The reversed number is: " + result);

```

Simple Calculator Program

```

import java.util.Scanner;

public class Calculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int ans = 0;

```

```

while (true) {
    System.out.print("Enter the operator: ");
    char op = sc.next().trim().charAt(0);
    if (op == '+' || op == '-' || op == '*' || op == '/' || op ==
'%') {

        System.out.println("Enter two numbers: ");
        int N1 = sc.nextInt();
        int N2 = sc.nextInt();

        if (op == '+') {
            ans = N1 + N2;
        }
        if (op == '-') {
            ans = N1 - N2;
        }
        if (op == '*') {
            ans = N1 * N2;
        }
        if (op == '/') {
            if (N2 != 0) {
                ans = N1 / N2;
            }

            if (op == '%') {
                if (N2 != 0) {
                    ans = N1 % N2;
                }
            }
        }

    }

    else if (op == 'x' || op == 'X') {
        System.out.println("Ending Calculator Program.");
        break;
    }

    else {
        System.out.println("Invalid Operator!!!");
    }

    System.out.println("The answer is: " + ans);
}

```

}

}