EAST WEST UNIVERSITY

Project Report

On

Store Program Category, Program name, Show Selected Program Name Converting .C File into .Java File and File Path in Database .

Submitted To:
Instructor: Md Mostofa Rassel
Assistant Professor , Department of Computer Science and Engineering
Course Tittle: Object Oriented Programming
Course Code: CSE110, sec-01

Submitted By:

Name:	ID:
Md.Yeasir Arafath Emon	2020-2-60-182
Nowshin Tabassum Ushna	2020-2-60-083

Introduction:

Project Aims and Objective:

The aims and objectives are as follows:

- Converting C code file into Java file
- A search column (list of Program) to search C and Java code file.
- We have presented a general model for complete C to Java conversion.

Responsibility Info:

- Database
- GUI, Programs

System Analysis:

In this part, we will discuss and analysis about the process of converting C code with .C file into Java code with .java file. C code and Java are quite different and would need to rework it substantially. Start by learning Java, which is an object oriented, as opposed to C's procedure oriented, style. In the list of program column (left side), first start with basic programs (3). For example,

Below is a program to reverse the case of input Character. Here in C, isLower() is system defined function under **Ctype.h** header file to check if the character is in lower or not.

toupper() converts the input parameter into equivalent uppercase char. In conversion to Java code, We should have do the following topic to understand this program –

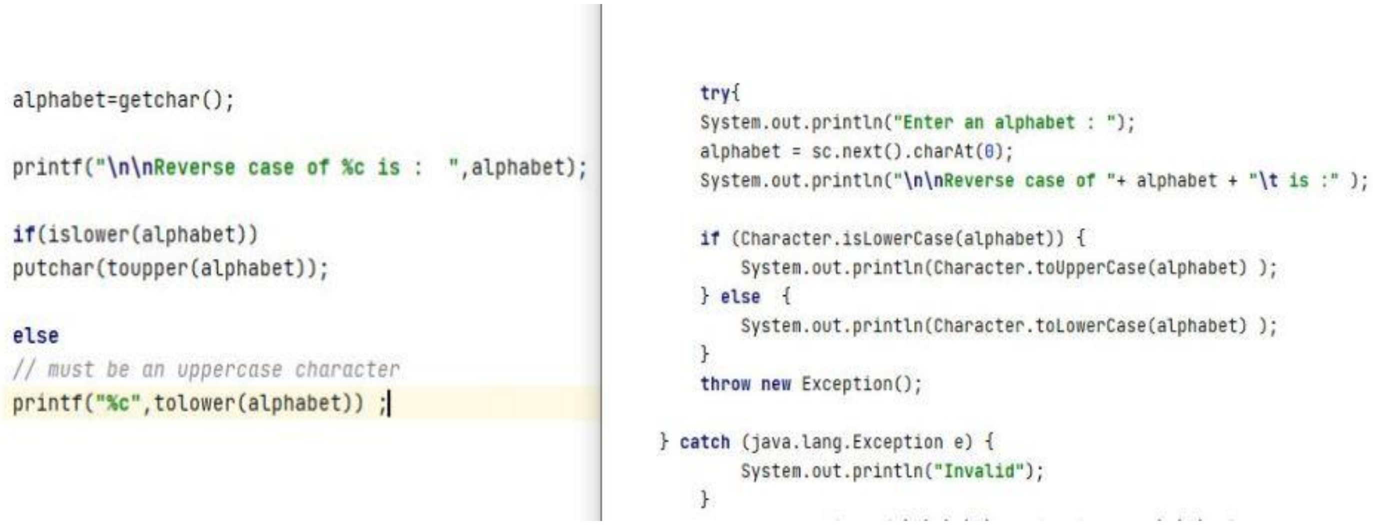
- Java (**java.util.Scanner)** package

- Java **main()** method
- Java **String** functions
- Java **System.out.println()** function

In Java, we use scanner class supports **next.charAt(0)**.To read char, we use it. **next()** function returns the next word in the input as a string and **charAt(0)** function returns the first character in that string.

Even, We use **Exception handling**.

The catch block is used to catch the exception thrown by statements in the try block. If the input is unexpected, the try catch block throw exception and print invalid in this code.



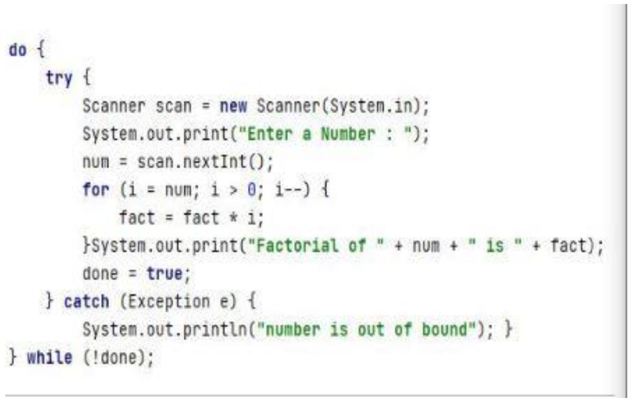
```c
alphabet=getchar();

printf("\n\nReverse case of %c is :  ",alphabet);

if(islower(alphabet))
putchar(toupper(alphabet));

else
// must be an uppercase character
printf("%c",tolower(alphabet)) ;
```

```java
    try{
    System.out.println("Enter an alphabet : ");
    alphabet = sc.next().charAt(0);
    System.out.println("\n\nReverse case of "+ alphabet + "\t is :" );

    if (Character.isLowerCase(alphabet)) {
        System.out.println(Character.toUpperCase(alphabet) );
    } else  {
        System.out.println(Character.toLowerCase(alphabet) );
    }
    throw new Exception();

} catch (java.lang.Exception e) {
        System.out.println("Invalid");
    }
```

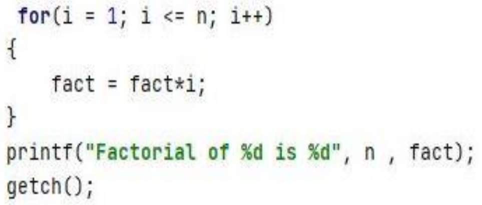
Fig: Converting C to Java Program

In loops programs, every loops consists of three parts Initialization, Condition, Updating. We do for loop, while loop and do while loop. In this converting C to Java, the loop's block same. Whenever an exception occurred in a loop the control gets out of the loop, by handling the exception the statements after the catch block in the method will get executed. But, the loop breaks. Even, we use do-while loop, it can check if the entered input was valid and simply stop the loop. In a do-while loop, by using **done**, the loop will run forever, unless no exception is thrown by the constructor. This makes the code more generic.

Example:



```
do {
    try {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter a Number : ");
        num = scan.nextInt();
        for (i = num; i > 0; i--) {
            fact = fact * i;
        }System.out.print("Factorial of " + num + " is " + fact);
        done = true;
    } catch (Exception e) {
        System.out.println("number is out of bound"); }
} while (!done);
```

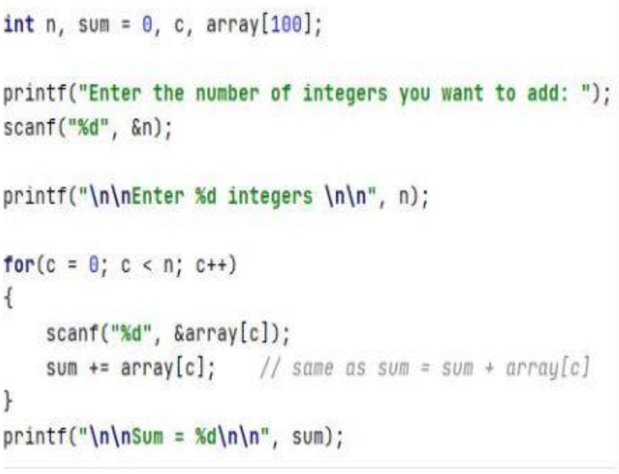


```
 for(i = 1; i <= n; i++)
{
    fact = fact*i;
}
printf("Factorial of %d is %d", n , fact);
getch();
```

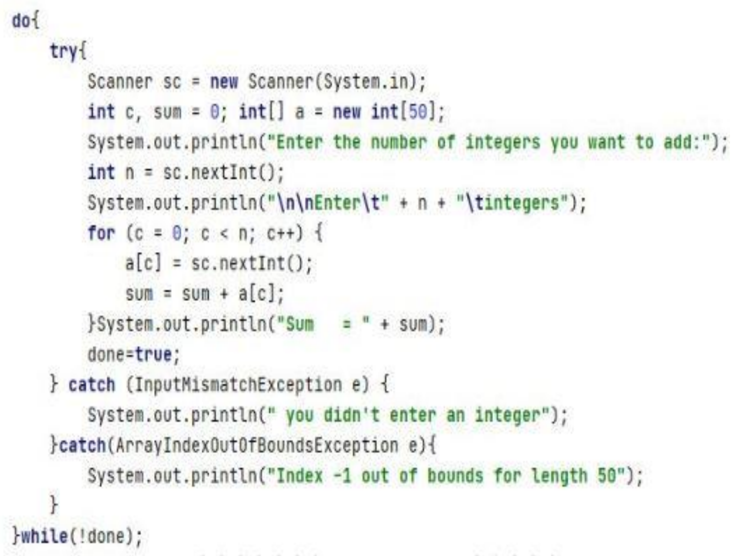
Java Program

C Program

Converting C code to Java in array, the main difference is declaration and initialization of array. Example: Let input array is an integer array having N elements. Declare an integer variable 'sum' and initialize it to 0. We will use 'sum' variable to store sum of elements of array. Using for-each loop, we will input array and add each element to sum variable. After termination of for each loop sum variable will contain the sum of all array elements. Whenever input is an (-ve) value or, the value greater than to the size of the array, then the **ArrayIndexOutOfBoundsException** is thrown. So, handle this exception using try catch. We use do-while loop, the loop will run forever, unless no exception is thrown by the constructor.



```
int n, sum = 0, c, array[100];

printf("Enter the number of integers you want to add: ");
scanf("%d", &n);

printf("\n\nEnter %d integers \n\n", n);

for(c = 0; c < n; c++)
{
    scanf("%d", &array[c]);
    sum += array[c];    // same as sum = sum + array[c]
}
printf("\n\nSum = %d\n\n", sum);
```



```
do{
    try{
        Scanner sc = new Scanner(System.in);
        int c, sum = 0; int[] a = new int[50];
        System.out.println("Enter the number of integers you want to add:");
        int n = sc.nextInt();
        System.out.println("\n\nEnter\t" + n + "\tintegers");
        for (c = 0; c < n; c++) {
            a[c] = sc.nextInt();
            sum = sum + a[c];
        }System.out.println("Sum   = " + sum);
        done=true;
    } catch (InputMismatchException e) {
        System.out.println(" you didn't enter an integer");
    }catch(ArrayIndexOutOfBoundsException e){
        System.out.println("Index -1 out of bounds for length 50");
    }
}while(!done);
```

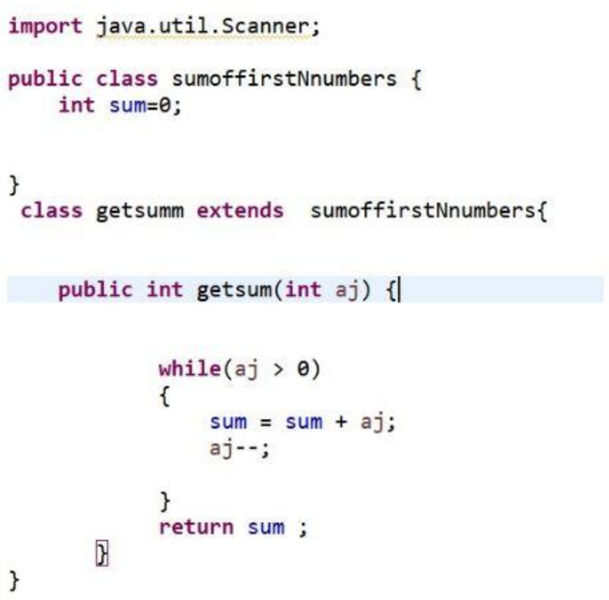
C Program

Java Program

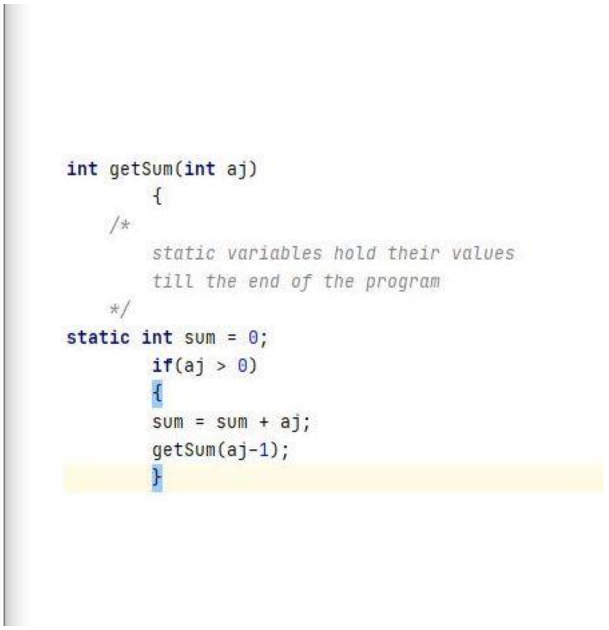
A function is Java can call itself. Such calling of function is called recursion. First Create an instance of the scanner class. Call recursion function to calculate the sum. We use **inheritance** to convert java program.

In java, when an “Is-A” relationship exists between two classes, we use inheritance. The supper class **sumOffirstNumbers ,** subclass is **getsum.**The keyword extends is used by the getsum to inherit the features of sumOffirstNumber.

Below the codes:

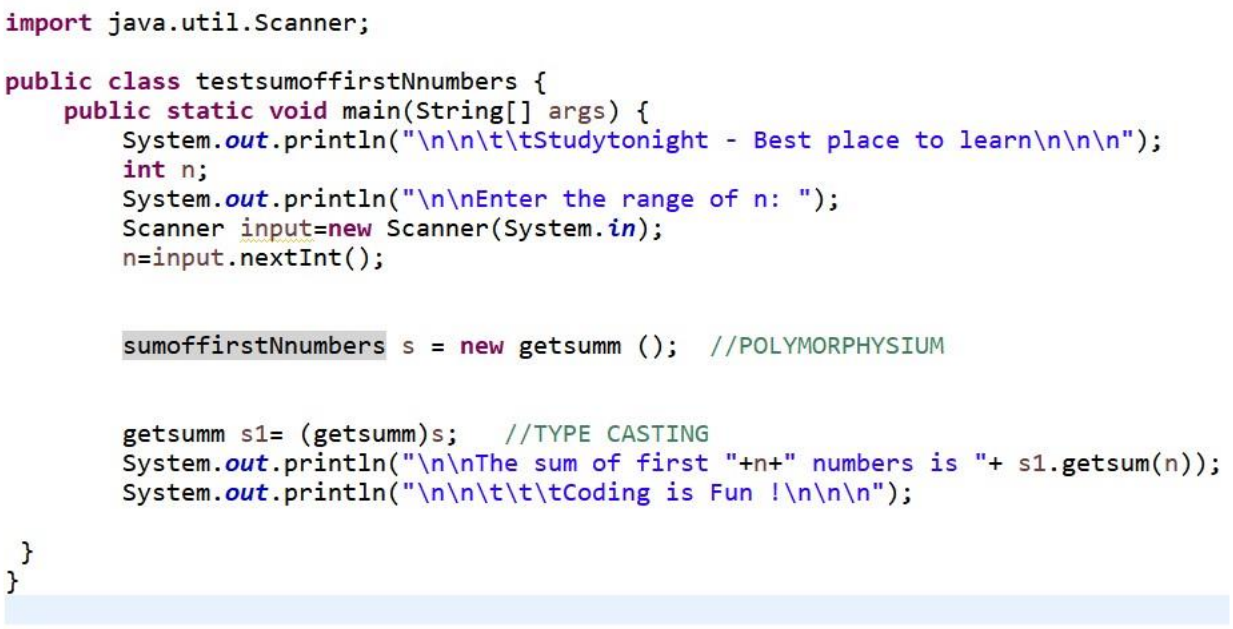


```
import java.util.Scanner;

public class sumoffirstNnumbers {
    int sum=0;


}
 class getsumm extends  sumoffirstNnumbers{


    public int getsum(int aj) {


            while(aj > 0)
            {
                sum = sum + aj;
                aj--;

            }
            return sum ;
        }
}
```



```
int getSum(int aj)
        {
    /*
        static variables hold their values
        till the end of the program
    */
static int sum = 0;
        if(aj > 0)
        {
        sum = sum + aj;
        getSum(aj-1);
        }
```

We using polymorphysium in converting java code.



```
import java.util.Scanner;

public class testsumoffirstNnumbers {
    public static void main(String[] args) {
        System.out.println("\n\n\t\tStudytonight - Best place to learn\n\n\n");
        int n;
        System.out.println("\n\nEnter the range of n: ");
        Scanner input=new Scanner(System.in);
        n=input.nextInt();


        sumoffirstNnumbers s = new getsumm ();  //POLYMORPHYSIUM


        getsumm s1= (getsumm)s;   //TYPE CASTING
        System.out.println("\n\nThe sum of first "+n+" numbers is "+ s1.getsum(n));
        System.out.println("\n\n\t\t\tCoding is Fun !\n\n\n");

 }
}
```