Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian

Associate Professor

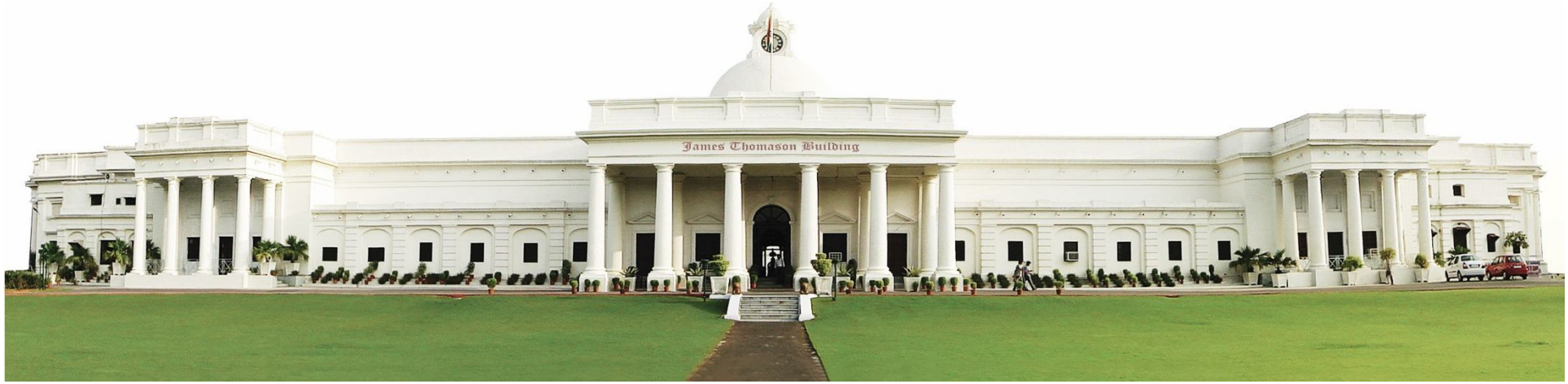
Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

Roorkee 247 667

balarfcs@iitr.ac.in

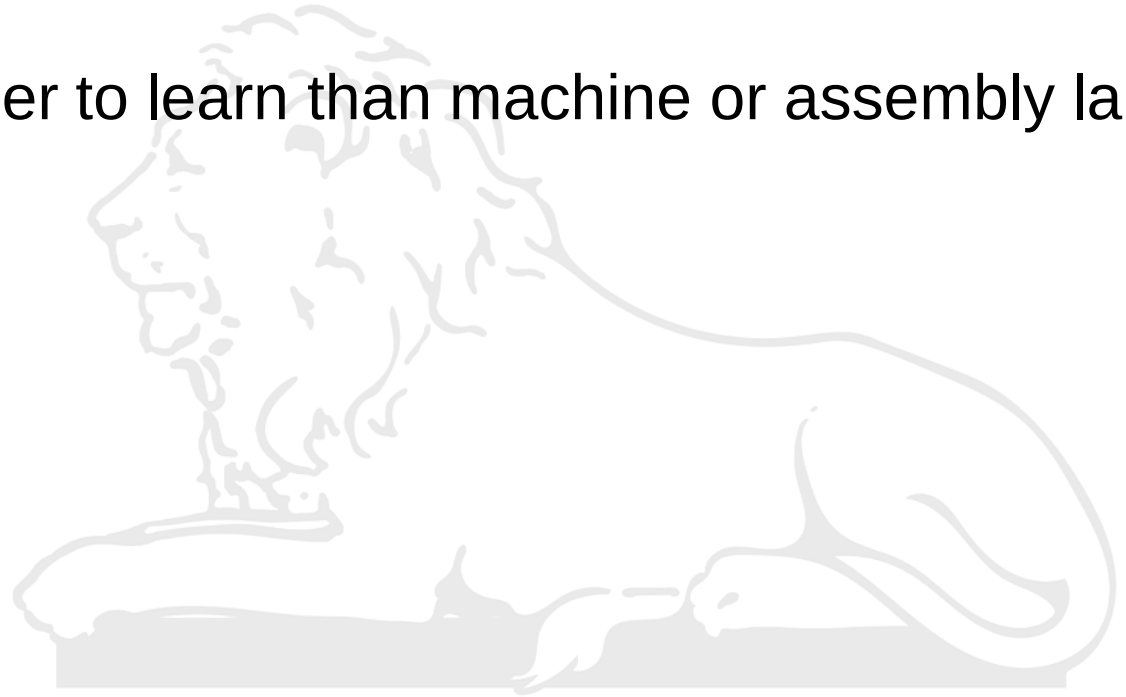
https://sites.google.com/site/balaiitr/



High-Level languages

- Instructions are quite English-like, and a single instruction can be written to correspond to many operations at the machine level.

- Easier to learn than machine or assembly languages.



Examples

- COBOL – developed in the 1960s for business transactions.
- FORTRAN – developed for mathematical calculations.
- Pascal - is a structured high-level language.
- C – is designed to support only procedure-oriented programming. Popular language for developing system applications such as operating system and compilers.
- C++ - is extension of C programming language that support object oriented programming and procedure-oriented approach.
- Java – is an object-oriented language.

Algorithm

- A set of **explicit, unambiguous finite steps**, which when carried out for a given set of **initial condition** to produce the corresponding **output** and terminate in **finite time**.

❖ Write an algorithm to find the product of 3 numbers.

```
1) Input A, B, C
2) Prod = A * B * C
3) Print Prod
```

Algorithm

- Ex. 2: Write an algorithm to swap two numbers

```
Step 1: Start

step 2: read 'a' and 'b' values

step 3: interchange the values
        temp=a
        a=b
        b=temp

step 4: write a and b values

step 5: stop
```

Algorithm

- Ex. 3: Write an algorithm to swap two numbers without using temporary variable

a → 100
b → 50

```
Step 1: Start

step 2: read 'a' and 'b' values

step 3: interchange the values
        a=a+b
        b=a-b
        a=a-b

step 4: write a and b values

step 5: stop
```

a → 150
b → 100 ✓
a → 50 ✓

Flowcharts

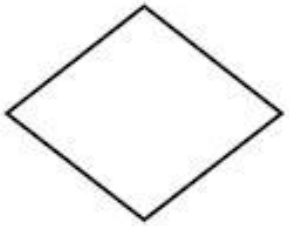
- Flowchart is a pictorial/graphical representation of an algorithm
- Uses symbols (boxes of different shapes) that have standardized meanings to denote different types of instructions
- Actual instructions are written inside boxes
- Boxes are connected by solid lines having arrow marks to indicate the exact sequence in which the instructions are to be executed

Basic Flowcharts Symbols

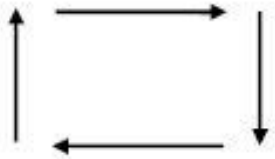
Terminal

Input/Output

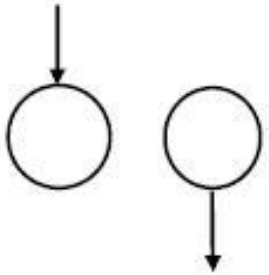
Processing



Decision

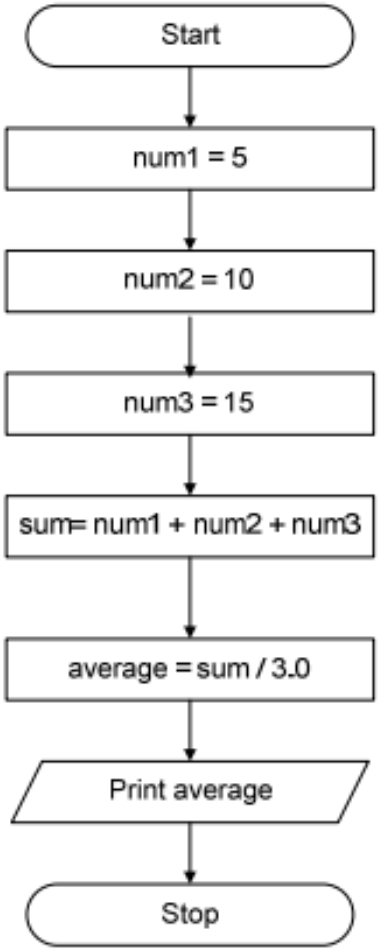


Flow lines



Connectors

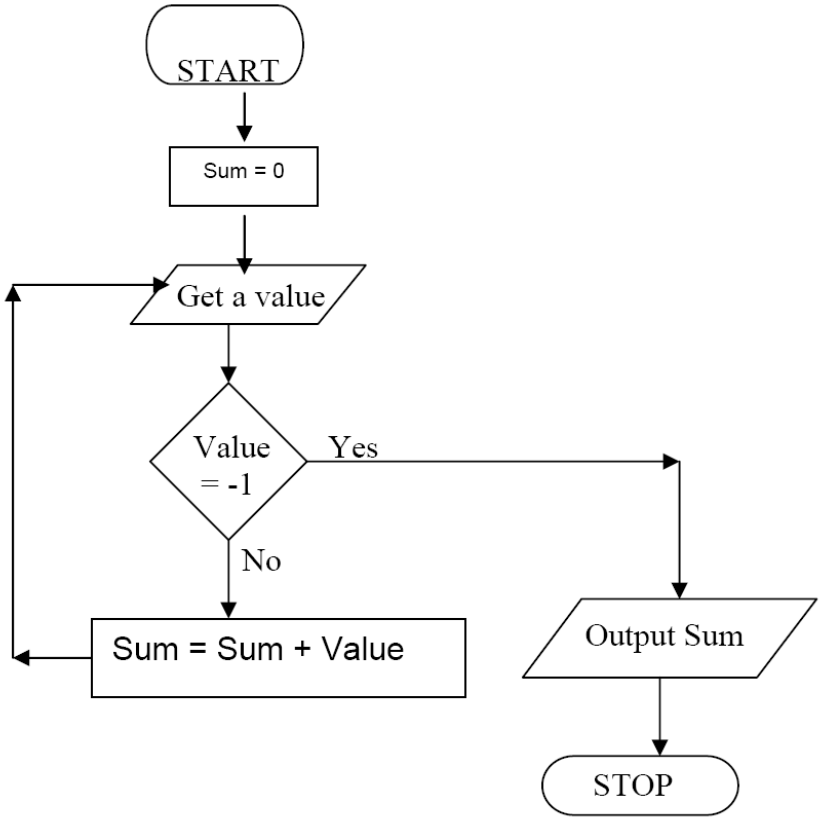
Flowchart of finding Average of three numbers



Flowchart 2

- Given list of numbers 28, 47, 492, 9387, 48960, 2, -1. Draw a flow chart to find the sum of this list.

OOAD



Pseudo code to find the average of 3 numbers

```
Declare Num1, Num2, and Num3 as integers

Declare Average as real

Write "Welcome to the program. Enter 3 positive
numbers",

INPUT Num1, Num2, Num3

Average = (Num1 + Num2 +Num3) / 3

Write "The average of three numbers entered is",
Average

Stop
```

Pseudo code for finding area and circumference of circle

```
INPUT "Enter radius of circle:", r

ac = 3.14 * r * r

c = 2 * 3.14 * r

PRINT "Area of circle=", ac

PRINT "Circumference of circle=", c

END.
```

Hello World Program in C++

C++

```
// Our first program in C++
#include <iostream>
using namespace std;
int main()
{
cout << “Welcome to Roorkee";
}
```

ideone.com

edit fork download

```
1.  #include <iostream>
2.  using namespace std;
3.
4.  int main() {
5.      // your code goes here
6.      cout<<"Welcome to IIT Roorkee";
7.      return 0;
8.  }
```

Success #stdin #stdout 0s 4536KB

stdin

```
Standard input is empty
```

stdout

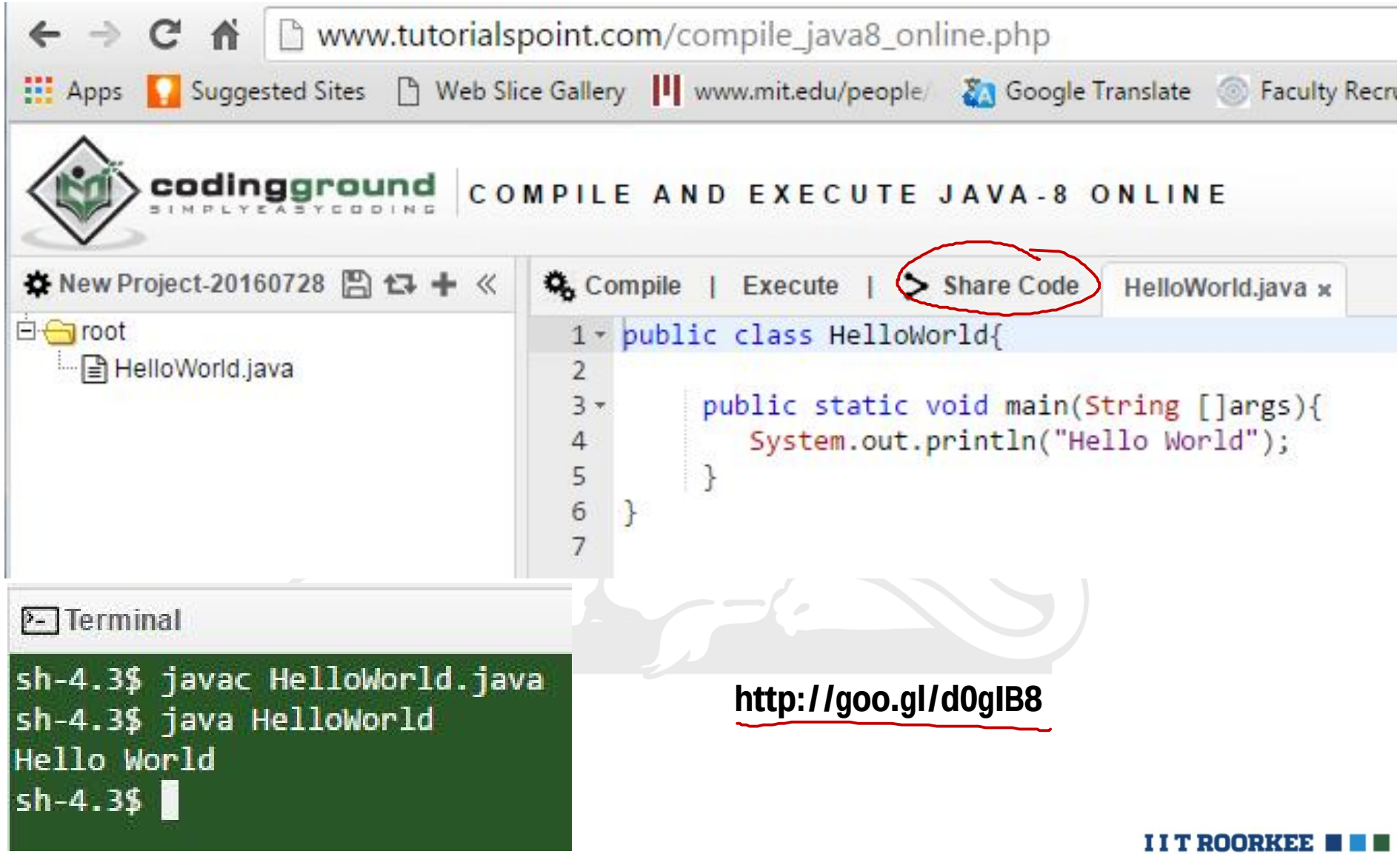
```
Welcome to IIT Roorkee
```

https://ideone.com/C1k5ei

Hello World Program in JAVA

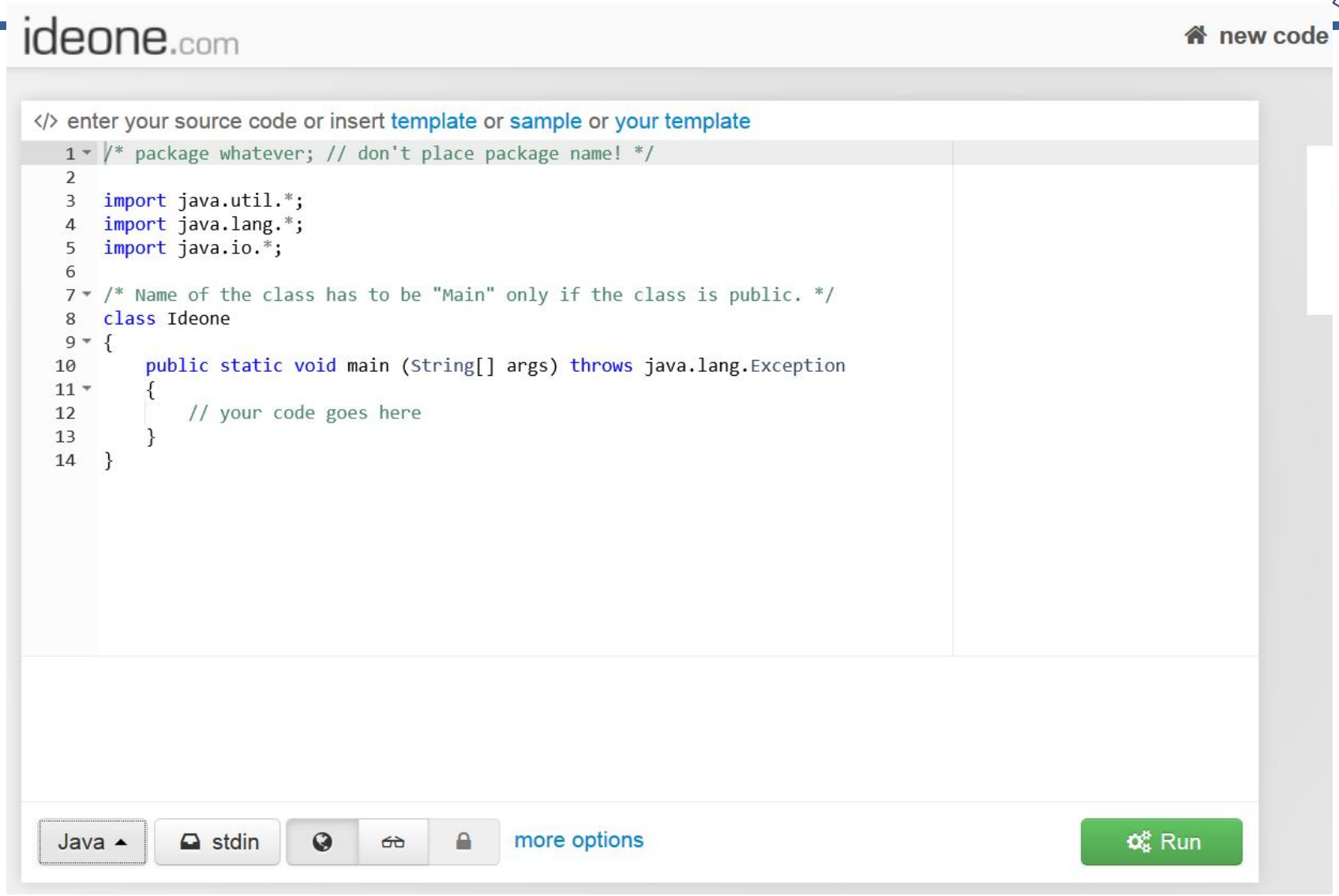
```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }

}
```

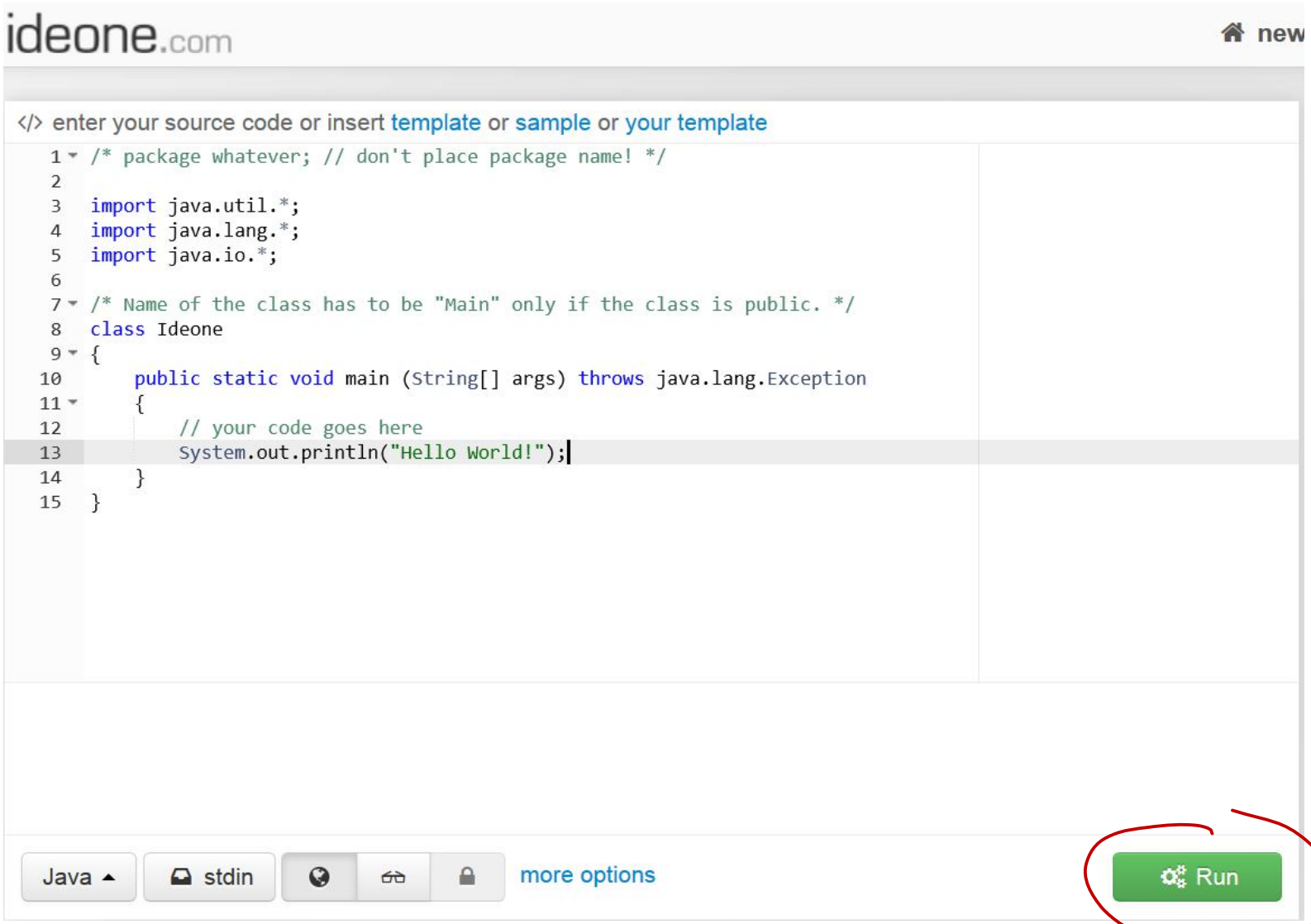
tutorialspoint.com (Online compilation)



http://goo.gl/d0gIB8

http://ideone.com/




ideone.com
new
enter your source code or insert template or sample or your template
/* package whatever; // don't place package name! */
import java.util.*;
import java.lang.*;
import java.io.*;
/* Name of the class has to be "Main" only if the class is public. */
class Ideone
{
public static void main (String[] args) throws java.lang.Exception
{
// your code goes here
System.out.println("Hello World!");
}
}
Java
stdin
more options
Run

stdin

```
Standard input is empty
```

stdout

```
Hello World!
```