Chapter-1 [Part-II]

Introducing BlueJ

Introduction: BlueJ is a Java development environment, designed and implemented by the BlueJ team at Monash University, Melbourne, Australia and the University of Southern Denmark, Odense. BlueJ is a window based platform for Java Development Kit (JDK). It requires installing JDK 1.3 version or more before installing BlueJ.

BlueJ is basically an IDE (Integrated Development Environment). It includes the following tools: an editor, a debugger and a viewer. BlueJ includes some tools that are helpful for creating Java applications. A tool of BlueJ IDE that is used for finding mistakes in a program is called debugger.

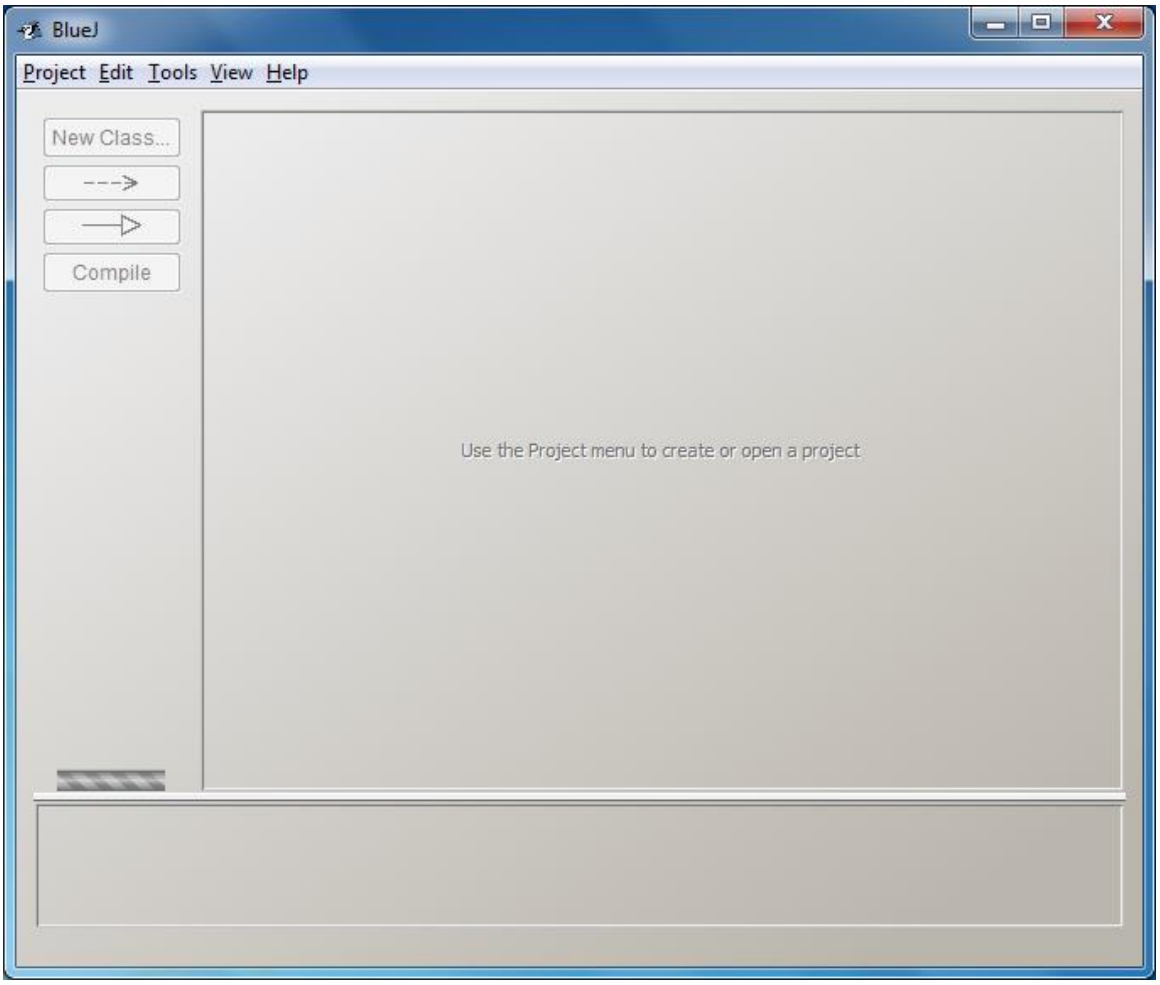
First you need to install the following to work on Java applications.

1. Latest BlueJ Software.
2. Latest JDK platform.

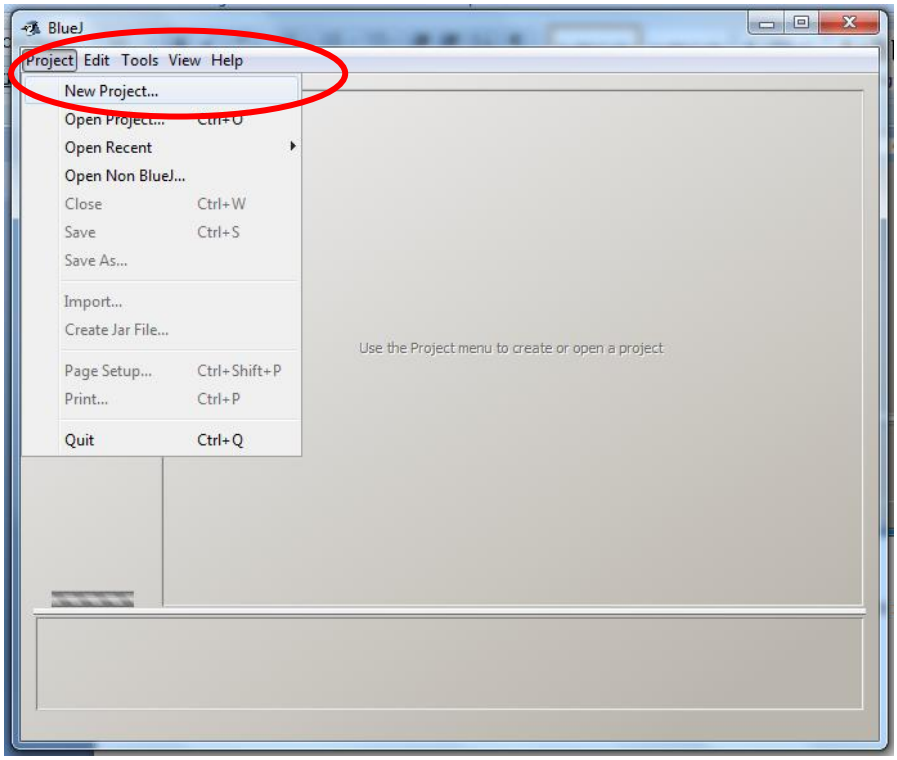
Creating the first Java program:

- Double click on the BlueJ icon present on the desktop screen.

- The following BlueJ window appears.

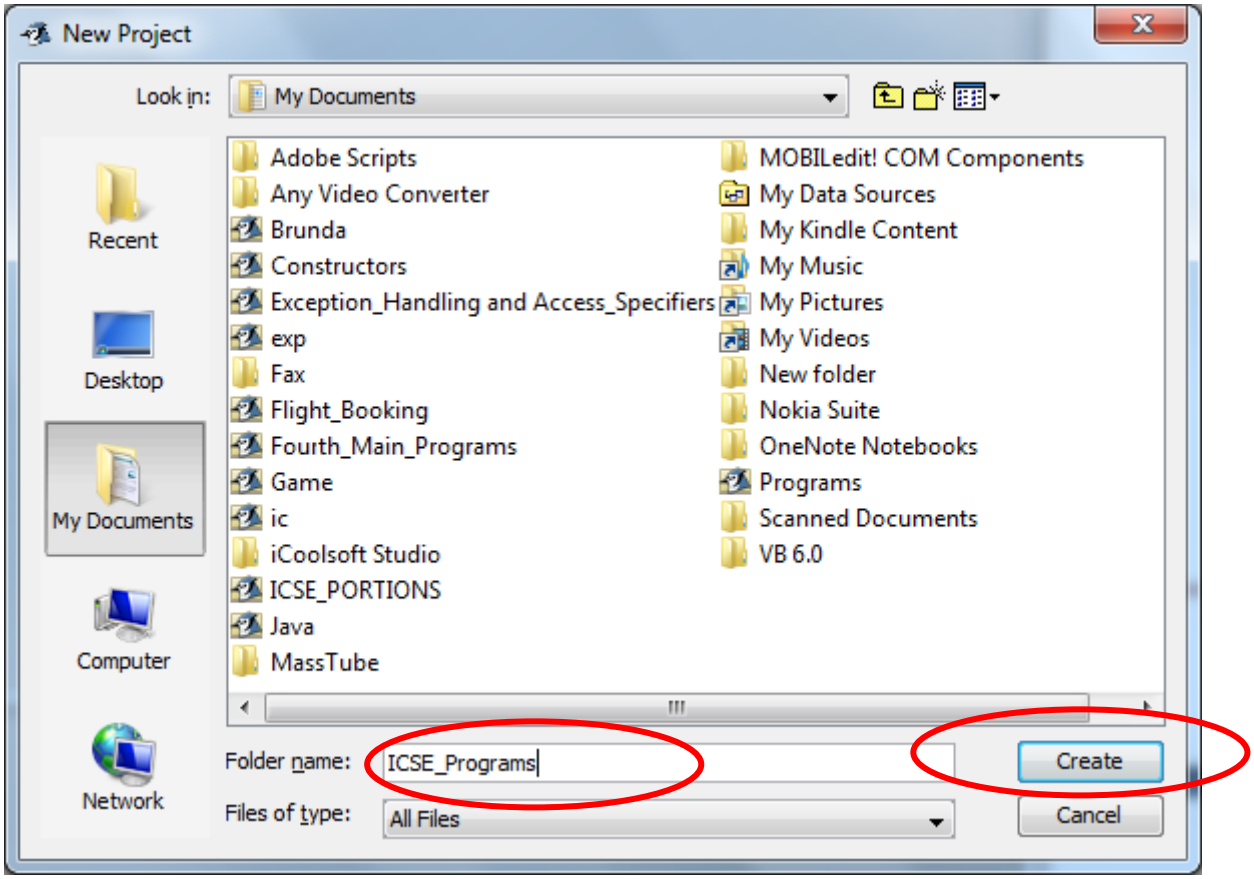


- Click on Project Menu and choose the option New Project as shown in the figure.

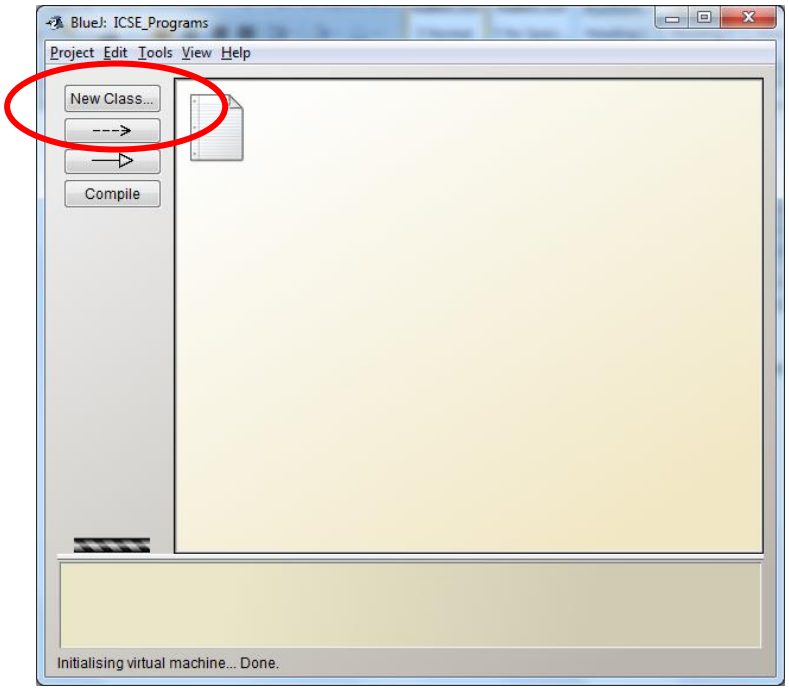


B. Ranganath, 9743602049

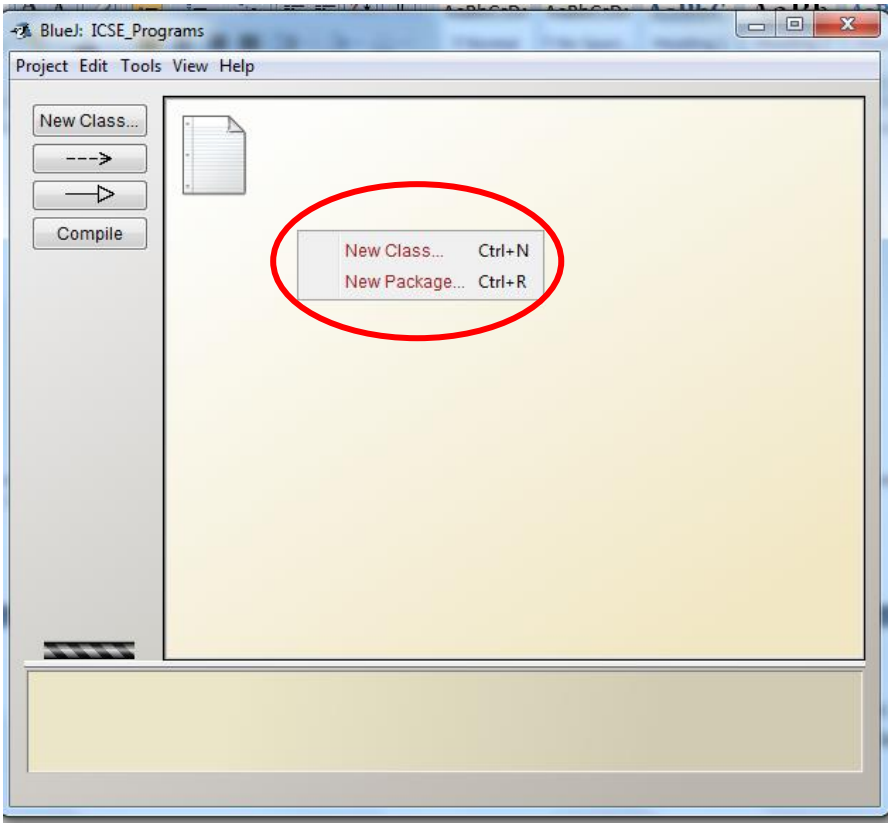
- Once you click on New Project, it will ask you give a name for the project and create in the appropriate directory wherever you require and click on create button.



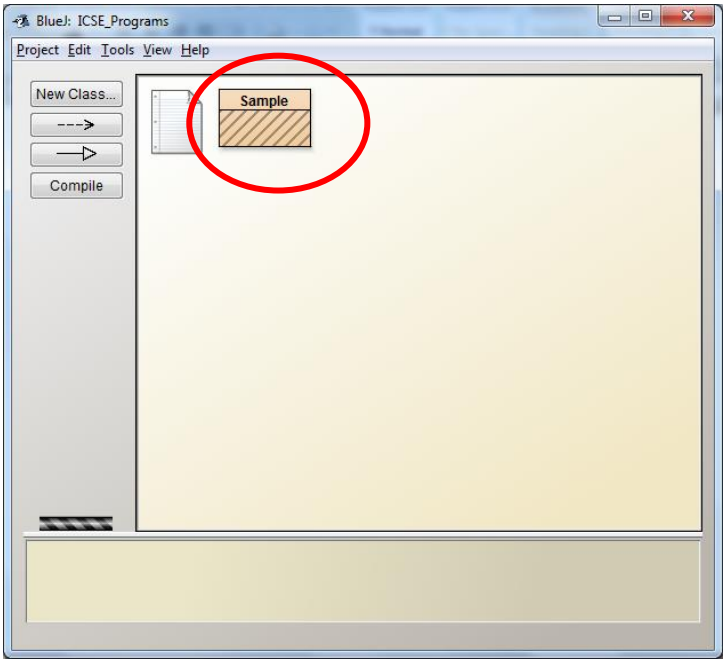
- Now you have created the new project with name ICSE_Programs. see the following window after creating the new project.

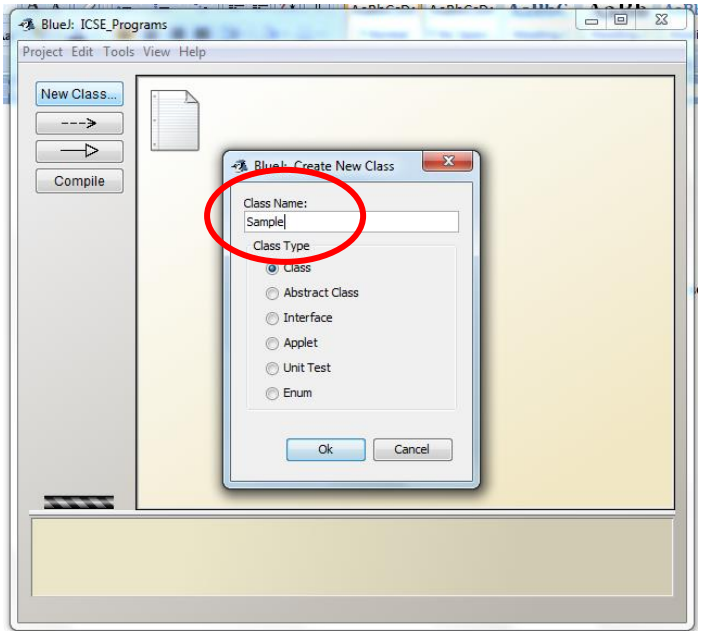


- Now to begin the java coding, you need to click on New Class button from left side of the window or right-click on the empty space and click on the option New Class.



- After clicking on New Class Option a dialog box appears to enter the name of a class and click on Ok. Here the class name given is Sample. see figure on the right indicating the icon of the java program.

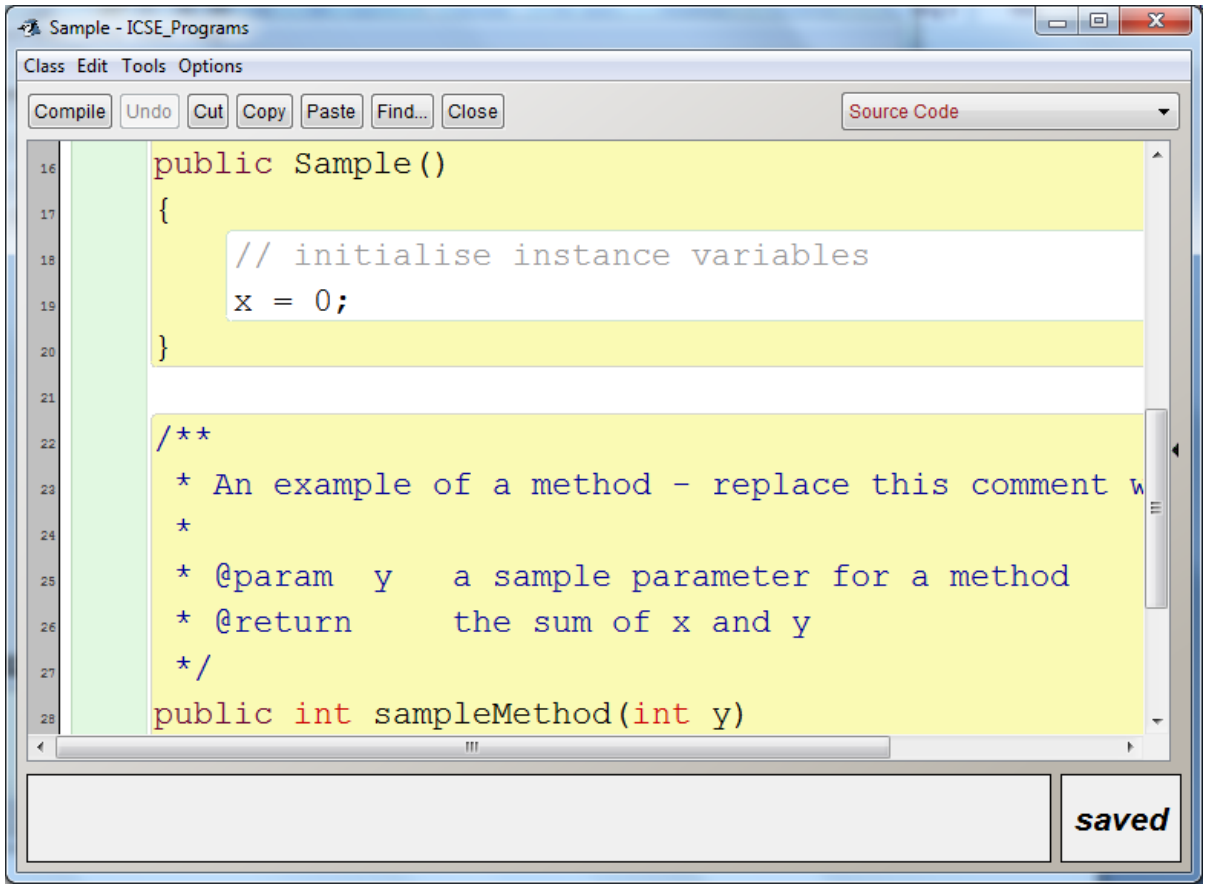




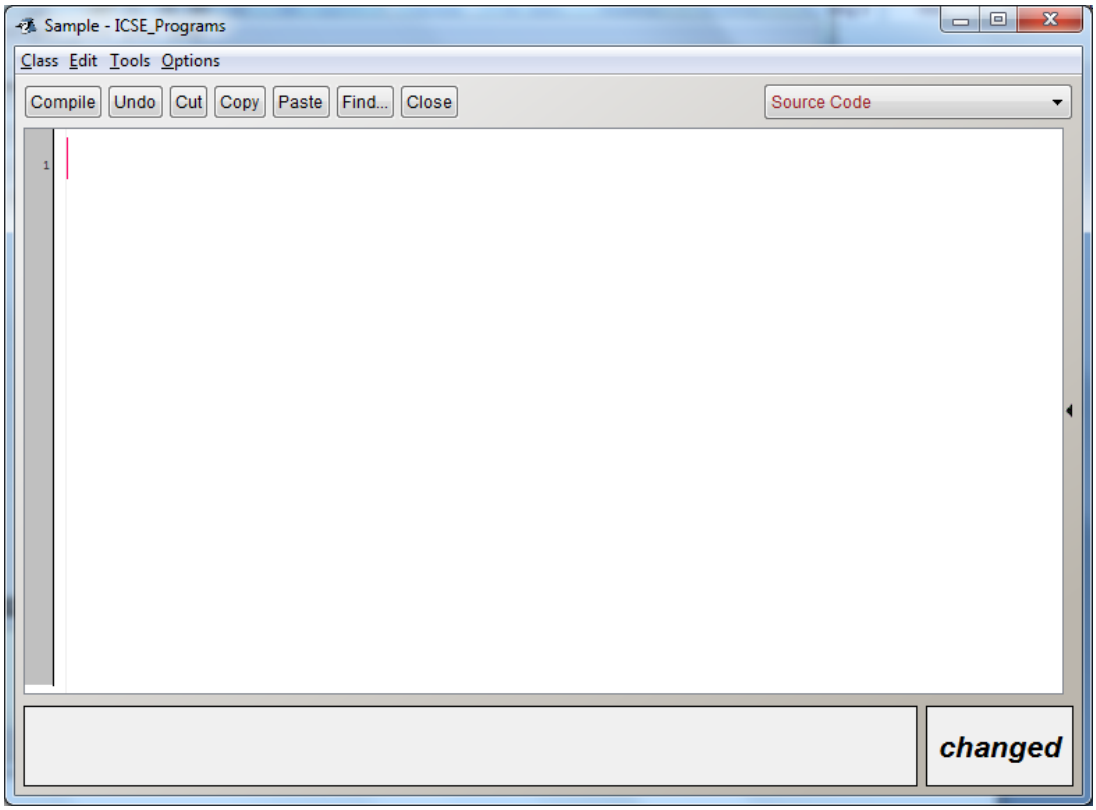
The Class name cannot begin with a number, or any other special character other than $ and _ (Underscore).

B. Ranganath, 9743602049

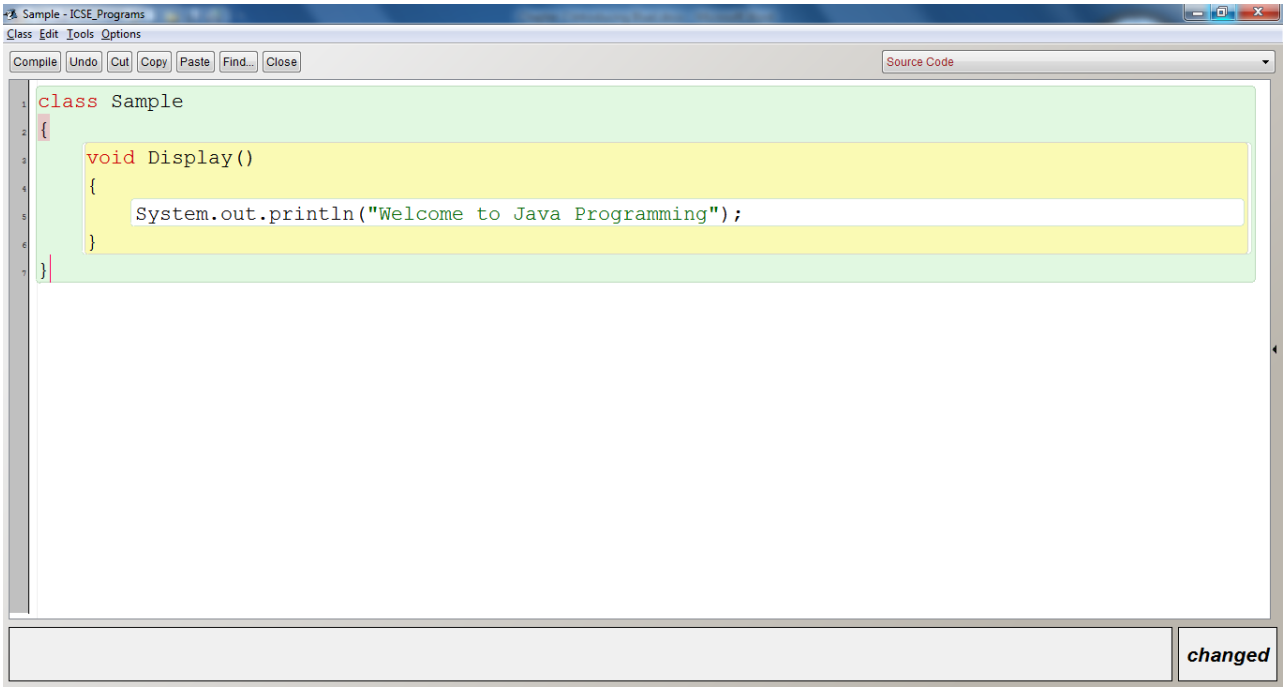
- To begin coding (writing Java Program), Double-click on the icon, the following Editor screen appears with a template.



- Select All and clear the template coding, the screen looks as shown



- Now begin typing your program, see the structure of a Java Program in the following screen.



To increase the font size click on options→Preferences and type the size of font you require to type the coding.

Explanation of the above Program:

Structure of a Java Program

Line No 1----------class Sample
Line No 2----------{
Line No 3---------- void display ()
Line No 4----------{
Line No 5---------- System.out.println ("Welcome to Java Programming");
Line No 6----------}
Line No 7----------}

Line No 1: Every java program should always start with the declaration of a class using the keyword class followed by a space and the name of a class.

Line No 2: Opening curly braces which indicates the starting of a class.

Line No3: It is a method/function within the class to perform any kind of operation. This method written using void keyword indicates that the method display () is not going to return any information. Any number of methods can be defined in the class. In the above example there is only one method defined. Always the name the name of the method/function should follow ( )-parenthesis.

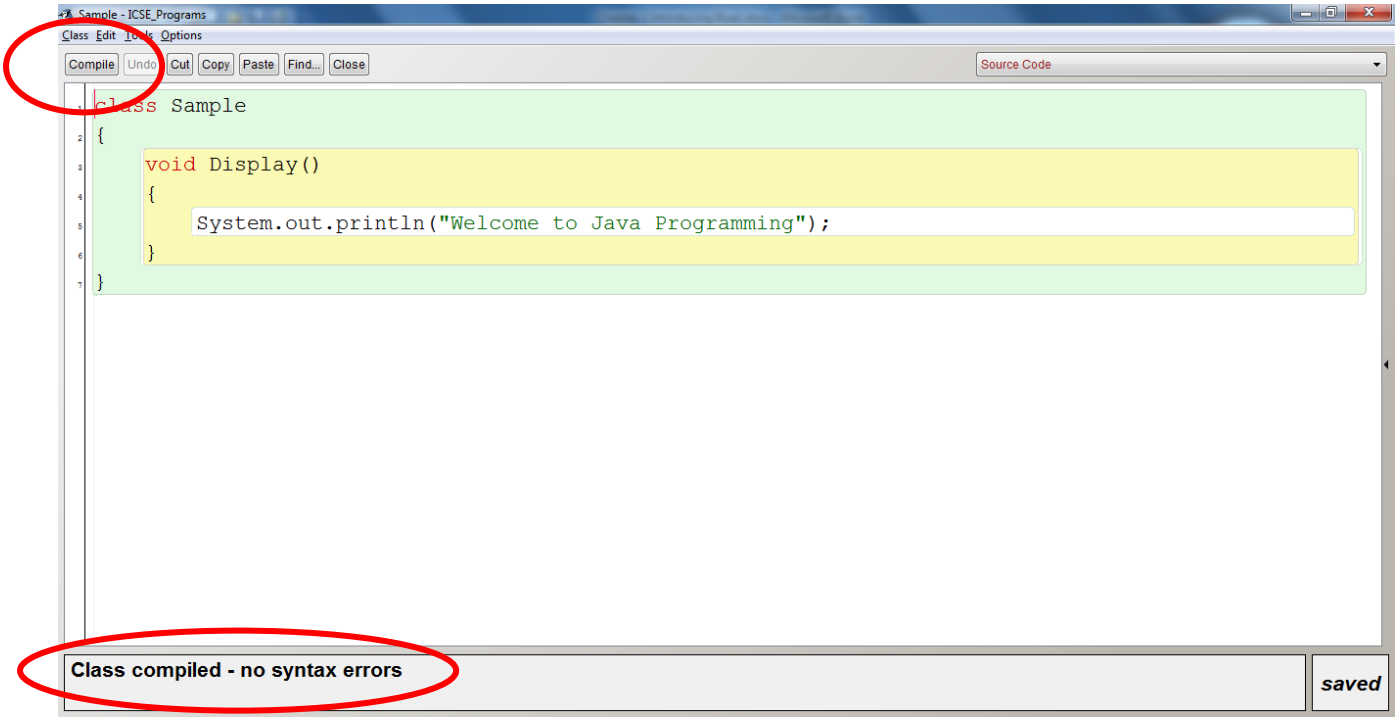
Line No 4: Opening curly braces which indicates the starting of a method/function.

Line No.5: In this System is a pre-defined class which contains the object out referring to the output device (i.e., Monitor) to display the information using the predefined method print () or println (). Print () function displays the information on the terminal window and the cursor remains in the same line, whereas println () display the information and brings the cursor to the next line automatically. The dot (.) operator used between System.out.println () is used to separate class name→System with the object name →out and the function→println (). ; (semi-colon) is used to end the java statement.

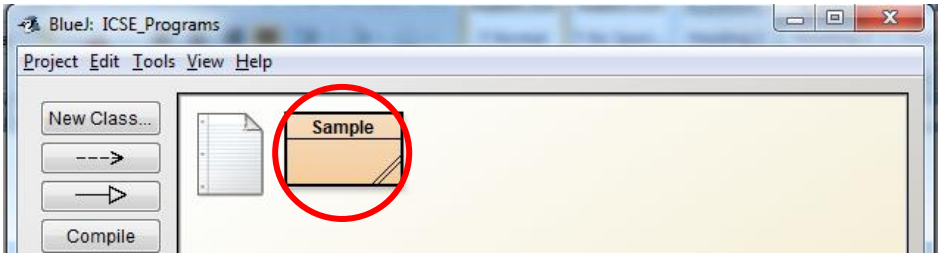
Line No 6: Closing curly braces which indicates the end of a function/method.

Line No 7: Closing curly braces which indicates the end of a class.

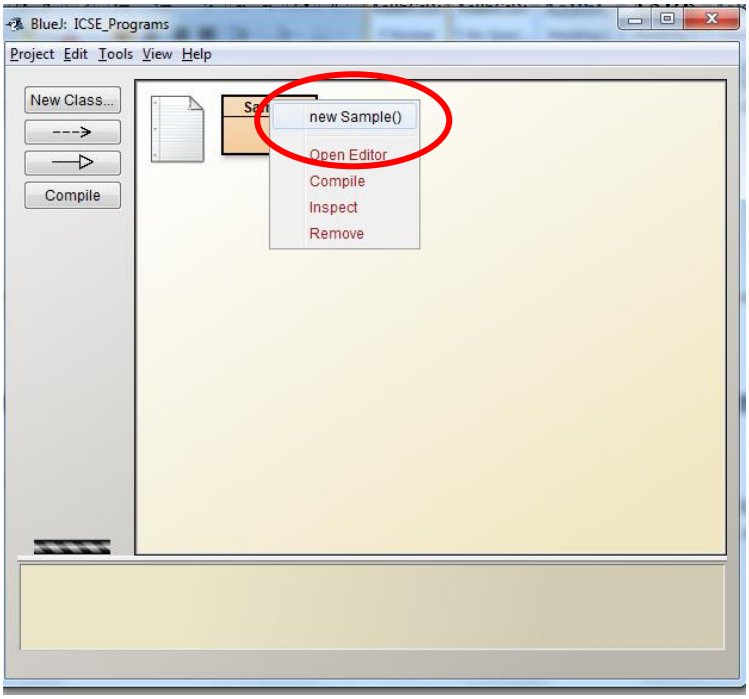
- After typing the complete program, the next step is to compile the program, which checks the entire code once and highlights if there is any syntax errors. If no errors, then it will display the message, Class Compiled-No Syntax errors below the screen.



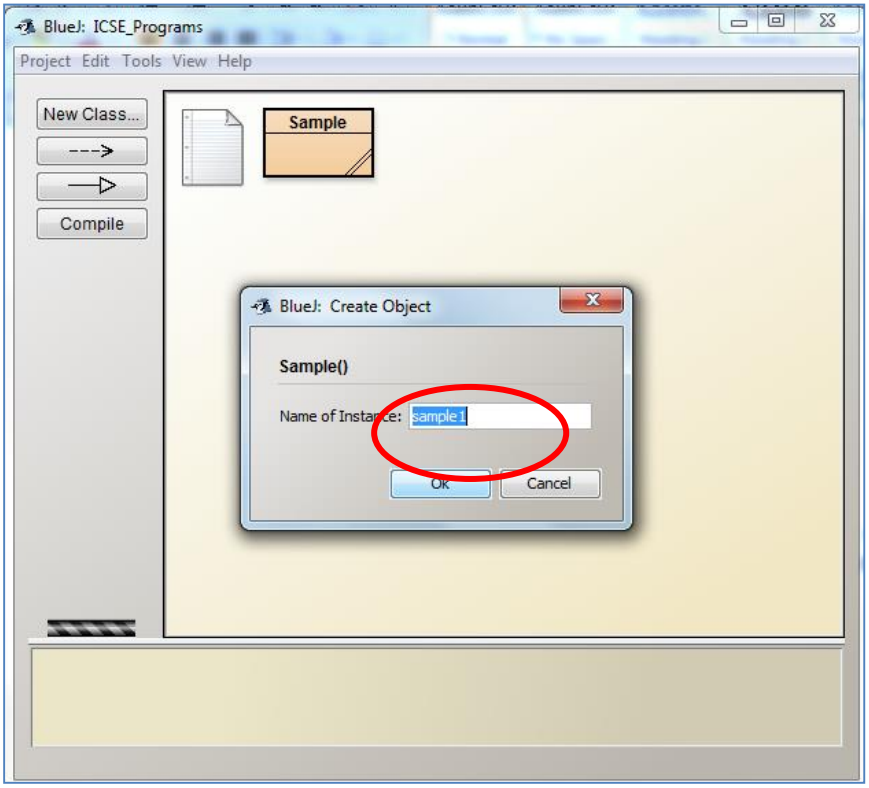
- Now the program is ready for execution. Look at the Icon in the screen.



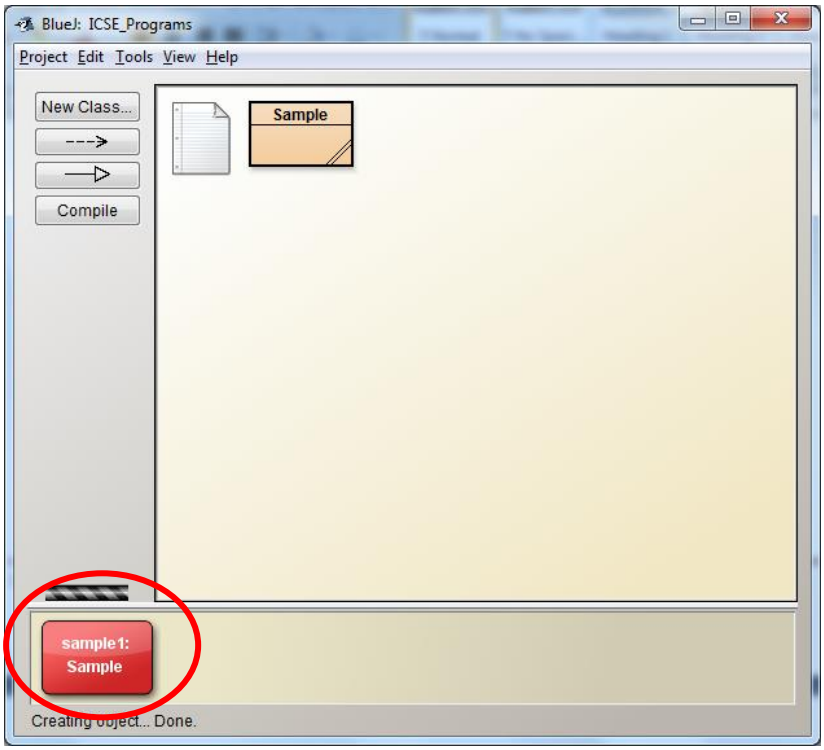
- To execute the program and see the output follow the given steps:-
  - First You need to create an object for the class Sample. To create an object, right-click on the Icon Sample and Select the option **new Sample()** as shown in the figure below:



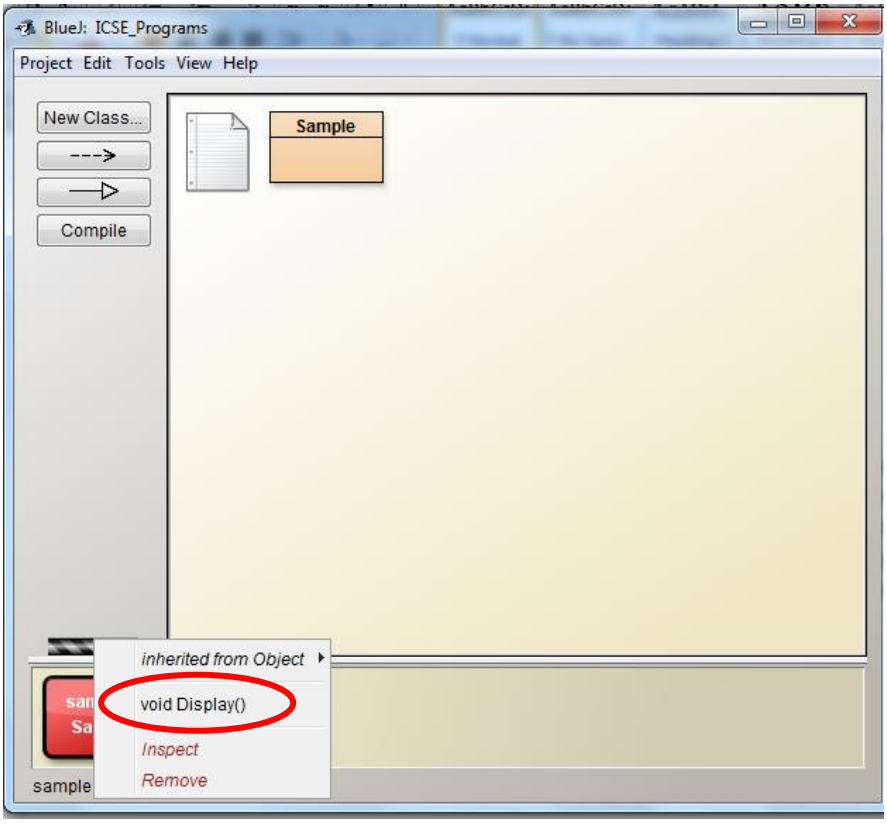
  - Once you click on new Sample(), a dialog box will be displayed asking the user to enter the name of object, you can either click on ok or type some other name for the object.



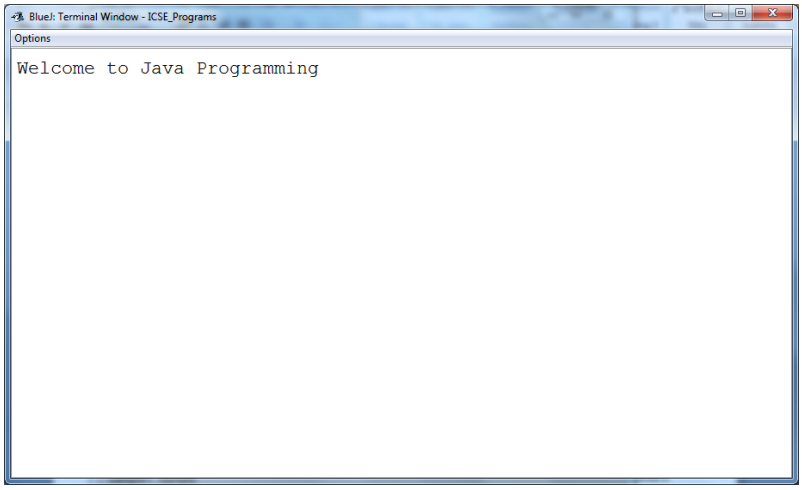
- Once you click on OK, the object gets created as shown in the figure.



- As you have learnt that every object has certain characteristics and behavior, in the above sample coding, we have only one behavior called Display() which performs an action of displaying the text message on the output screen.
- Now to display the output on the Terminal Window (Output Screen), right-click on the object and select the behavior(method) Display() as shown in the figure.



- After Clicking on void Display() method, the following output screen appears which displays the result of java programs.



print() method displays the output and the cursor remains in the same line, whereas println() method displays the output and the cursor moves to the next line for further display of outputs.

Using Escape Sequence characters \n and \t

If we want to display the output on different positions on the screen, then we can use \n and \t which are called as non-graphic characters to format the output.

\n - represents new line

\t - represents tab space

Example:

```
System.out.println ("Low-Level Language\nHigh-Level Language");
```

Output:

```
Low-Level Language
High-Level Language
```

In the above example, first it prints Low-Level Language, and then the cursor moves to the next line due to \n and prints High-Level Language.

What will be the Output of the following Code?

```
System.out.println ("Monitor\tPrinter\nMouse");
System.out.print ("\tKeyboard");
```

Points to remember

- When you compile a program written in the Java programming language, the compiler converts the human readable source file into platform independent code that a JVM can understand. This platform independent code is called byte code.
- Like any other programming language, we can use Java to write or create various types of computer applications. The word platform is new software platform designed to deliver and run highly interactive, dynamic and secure applications on networked computer systems.
- In ordinary compilation, the source code is converted to a machine code, which is dependent upon the machine or the platform. This resultant machine code is called native executable code.
- Contrary to ordinary compilers, the java compiler does not produce native executable code for a particular machine. Instead it produces a special format called byte code. The Java byte code looks a lot like machine language, but unlike machine language java byte code is exactly the same on every platform.
- The JVM-Java Virtual Machine is an abstract machine designed to be implemented on the top of existing processors. It hides the underlying operating system from Java applications. Programs written in Java are compiled into Java Byte code, which is then interpreted by a special Java Interpreter for specific platform. Actually this Java interpreter is known as the Java Virtual Machine (JVM).
- There are two types of Java programs, Internet Applets and Stand alone applications.
- Internet Applets are small programs that are embedded in web pages and are run on the viewers machine in a secured manner.
- Stand alone application is generally a software application that does not require web browser to run.
- JDK (Java Development Kit) contains necessary tools to execute and develop java programs. Some of the tools are javac→java compiler, java→ java interpreter, javah→includes header files(packages), jdb→ java debugger etc.
- The Java programs needs to be written just once, which can be run on different platforms without making changes in the java program. Only the Java interpreter is changed depending upon the platform. This characteristic is known as **WORA(Write Once Run Anywhere)**.
- Other Escape sequence characters are \" , \\, \b, \r

Practical Exercise

1. Write a Java Program to display your Name, class &Sec, School in different lines.

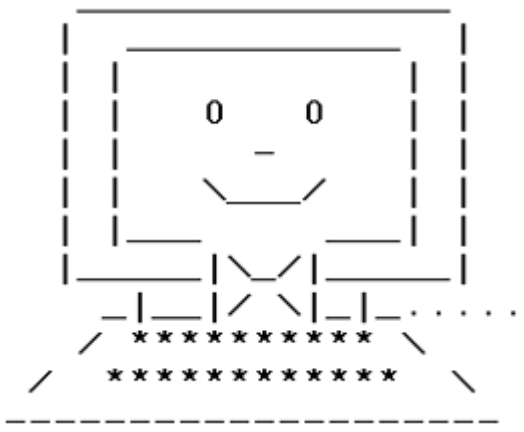
2. Write a Java Program to display the following pattern using only 3 print statements.

```
        I
        N
I       N       D       I       A
        I
        A
```

3. Write a Java Program to display the following pattern using only 3 print statements.

```
P                               M
        R               A
                O       R
                        G
                O       R
        R               A
P                               M
```

4. Write a Program to display the following Output:



5. Write a Java Program to display any 5 proverbs in different lines along with " ".

Write the Output of the following Statements:-

1. System.out.println ("This is Line1\nThis is line2");

2. System.out.print ("Class ");

   System.out.print ("Object");

   System.out.println (" are the OOP concepts");

3. System.out.println ("\\\\//\\\\//");

4. System.out.println ("\"Platform Independent\"");

5. System.out.println ("A\tB\tC\tD\nE\tF\tG\nH\tI\nJ");