How to Create a Java program

Mateen Ahmed Abbasi
Department of Computer Science
Khwaja Fareed University Of Engineering and Information Technology

WHAT YOU NEED

- A TEXT EDITOR WHERE YOU CAN TYPE YOUR PROGRAM AND SAVE THE FILE (GENERICALLY CALLED THE SOURCE CODE FILE).
  - I.E. NOTEPAD++ (AVAILABLE ON YOUR COMPUTER STATIONS)
- THE JAVA COMPILER TO *COMPILE* THE PROGRAM
- THE JAVA INTERPRETER TO *INTERPRET* (ALSO REFERRED TO AS *EXECUTE*) YOUR PROGRAM.

INFORMATION ON JAVA TOOLS

- JDK
  - Java Development Kit- (Used when you write a program)
  - Provides:
    - The compiler
    - Other tools (not needed in this class).
- JRE
  - Java Runtime Environment- Used when you are ready to execute (interpret) a program.
  - Provides:
    - The runtime environment JVM (Java Virtual Machine)
    - The interpreter

INFORMATION ON JAVA TOOLS

- Notepad++
  - The text editor tool used by this class to write (edit) java programs.
    - (note that there are other editors besides Notepad++ but this one is the approved for the class).

COMPILING AND RUNNING A JAVA PROGRAM

1. Use Notepad++ to write a program.
   – Use Java API as help.
   – Use text and text examples for help.
   – This file is sometimes referred as the “source code file”
2. Save the source code file with a name and extension .java

   i.e. MyFirstProgram.java

COMPLILING AND RUNNING A JAVA PROGRAM

- THE COMPILING AND THE RUNNING OF THE PROGRAM CAN BE DONE:

1. BY USING THE NOTEPAD++ TOOLBAR MENU UNDER MACRO-> COMPILER JAVA OR RUN JAVA OR
2. BY USING A DOS WINDOW AND TYPING DOS COMMANDS.

COMPILING AND RUNNING A JAVA PROGRAM USING A DOS WINDOW

3. Call the java compiler from the directory you have your source code file saved.
   - Open a DOS pane (we open a DOS window from accessories and clicking on “command prompt”)
   - On the DOS pane go to where your program is saved

     i.e. C:\COSC1102\MyPrograms\>

     where MyPrograms is the folder where you saved your program (we change the directory on the DOS pane going down by typing cd.. or going up one directory step by typing cd followed by the name of the folder one level up).
   - Call the java compiler, javac , to compile your program

     ```
     i.e. C:\COSC1102\MyPrograms\> javac  MyFirstProgram.java
     ```

COMPILING AND RUNNING A JAVA PROGRAM

4. The compiler will complain if errors were made in writing the source code. These errors are called: **“COMPILER ERRORS”.**
   - Correct source code errors as needed and re compile.
   - Keep this process going until no errors are reported by the compiler.
   - Compiler error messages identify the line # where the error was made and give some hint (usually not very well phrased-some guessing required gained with experience of using the compiler).

COMPILING AND RUNNING A JAVA PROGRAM

5. When all errors are corrected the compiler creates a file called the “bytecodes file” under the name you gave to the program but with the extension: .class

   i.e. MyFirstProgram.class

   – This file is created in the same folder as your source code file.
   – Opening this file does not give you any recognizable information by the human eye (No English words).

COMPILING AND RUNNING A JAVA PROGRAM

6. Now we are ready to interpret (or “execute” as the term used in other programming languages).
   - On a DOS pane call the interpreter (called java)

     i.e. C:\\MyPrograms\> java MyFirstProgram

     Notes:

     - java is the command for the interpreter.
     - Your bytecodes file name is used without any extension.
     - The output of the program will appear on the DOS window(pane). We could, however, get errors during execution, called: **“RUNTIME ERRORS”.**

-

COMPILING AND RUNNING A JAVA PROGRAM

7. Runtime Errors
   - These are errors that take place during the time the program runs (these errors can not be caught by the compiler).
   - Quite often they are referred to as “exceptions”
   - Usually the result of programmer’s mistakes.
   - They are reported by Java’ s runtime environment called JVM (Java Virtual Machine).
   - Need to go back to the source code file and correct them, then re compile and try to interpret again.
   - The interpreter will give you messages (usually cryptic) regarding the runtime errors. Sometimes we can make sense out of the message, but other times we have to kind of guess.

COMPILING AND RUNNING A JAVA PROGRAM

8. If there are no compiler or runtime errors, it is possible to get another category or errors called **<u>“LOGICAL ERRORS”</u>** (in spite of the fact that your program was inetrpreted)
   - It is possible that you are not getting any errors but your program does not give you the correct result.
   - This is due to your logical errors when you wrote the code:
     - i.e. Suppose that you meant to add but instead you multiplied two numbers.

STRUCTURE OF A JAVA PROGRAM

- Your java source code should have the following sections:
  - Import statements (not always needed)
  - The class name
  - “Global” variables declared (not always needed). Also called instance variables.
  - Methods inside the class
    - One of those methods must be the **main** method
      - Inside the method we have “local” variables declared and the source code for the method.

STRUCTURE OF A JAVA PROGRAM

- Note: A java program can consist of more than one class sometimes.
- Not all Java programs (classes) need a main method.
- But, in order for your program to be interpreted and produce a result then at least one of your classes needs to have a main method. The program starts always with the interpretation (execution) of the main method in whatever class is located.
- The classes without a method are there to provide additional support for the class that has the main method.

- Example of a simple program. This program has no import statements (not needed for this program).
- **public class MyFirstProgram**

```
{
        //double slashes are used for comments
        //anything after the double slashes is disregarded by the
compiler
        //the name of the class is: MyFirstProgram
        public static void main (String[] args)
        {
            System.out.println(“This is a Java program”);

              //the above statement tells the computer to print out on
the DOS window the String (text)           //enclosed by the double
quotation marks.
        }
 }
```

- Save the file as MyFirstProgram.java
- Notice that the name of the source code file has to match the name of the class
- The output of this program will appear on the DOS pane. What is the output?

STRUCTURE OF A JAVA PROGRAM

- <u>Observations</u>
  - A class starts with a curly bracket and ends with a curly bracket.
  - A method starts with a curly bracket and ends with a curly bracket
  - Comments that get disregarded by the compiler start with //
  - A line of code ends with semicolon ;
  - The output in English words in this case is surrounded by double quotations (this type of output is called : String).
  - System.out.println is a command that tells the computer’ s operating system that we want to output on the standard output DOS window as a String.
  - The code that ends with the semicolon is sometimes referred to as ”one line of code” or statment.

STRUCTURE OF A JAVA PROGRAM

- Class
  - Notice that the program starts with something called “ class” after the word public. A java program always needs a class with a name (chosen by you).
  - A class represents a category of items. We will explain the meaning of the class further as we go deeper into the course.
  - It is the basis of what is called Object Oriented Programming (OOP).

STRUCTURE OF A JAVA PROGRAM

- Method
  - The part of the java program where you place the code that performs the desired action(s) that bring the desired result(s). The method name in the previous example was “main”.
  - There can be more than one method in the program.
  - There must be a method called main besides any others (the other methods will have different names).
  - The main method is <u>always written </u>the way shown in the example.
    i.e public static void main(String[] args).

STRUCTURE OF A JAVA PROGRAM

- Notice that a line of code always ends with a semicolon.
  - System.out.println(“Hello World”);
- Also notice that Java is case sensitive for some words called “keywords”.
  - Keywords are reserved words by the language and are case sensitive.
  - i.e the word “class” is a keyword. Typing “Class” with a capital C will create an error by the compiler (because of the capital C).
  - public is also a keyword etc.
  - Names that you give to classes and methods and variables are not case sensitive but nevertheless have some restrictions (i.e you can not name a class with numbers).

STRUCTURE OF A JAVA PROGRAM

- <u>COMBINING TWO OUTPUT STATEMENTS by using the concatenation operator : +</u>
  - Output statements in the form of English are called Strings.
  - We can combine two Strings by using the plus operator +
  - When the plus sign is used in that context it is called the “concatenation operator”.
    - We will see that quite often in Java a symbol can be used in more than one context. (as with the + symbol where sometimes is used as an arithmetic plus and sometimes as a means to combine Strings).

STRUCTURE OF A JAVA PROGRAM

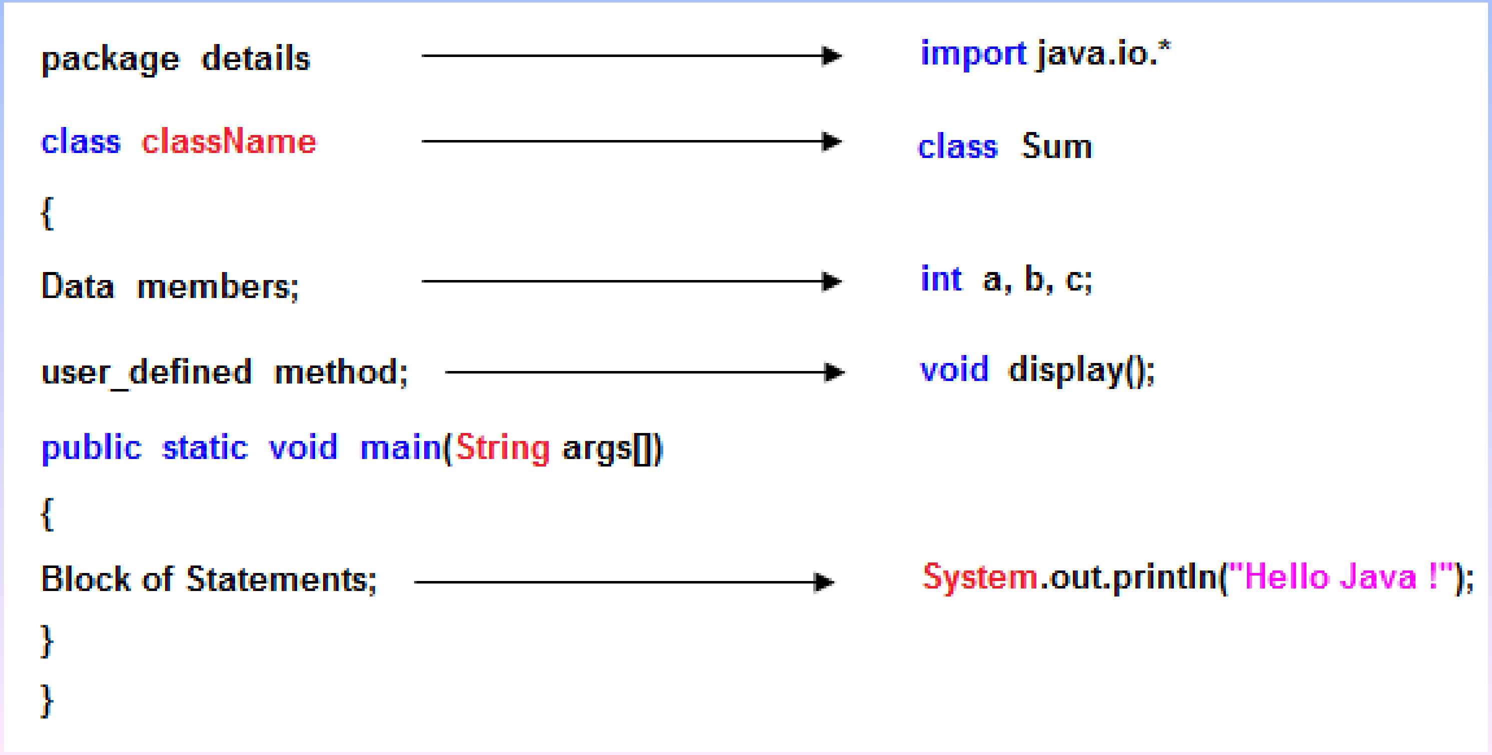
- Example

-
  ```
  public class CombineTwoStrings
  {
          public static void main(String[] args)
          {
                  System.out.println("This example combines"+"two Strings");
          }
  }
  ```

Structure of Java Program

Structure of Java Program

- Structure of a java program is the standard format released by Language developer to the Industry programmer.
- Sun Micro System has prescribed the following structure for the java programmers for developing java application.



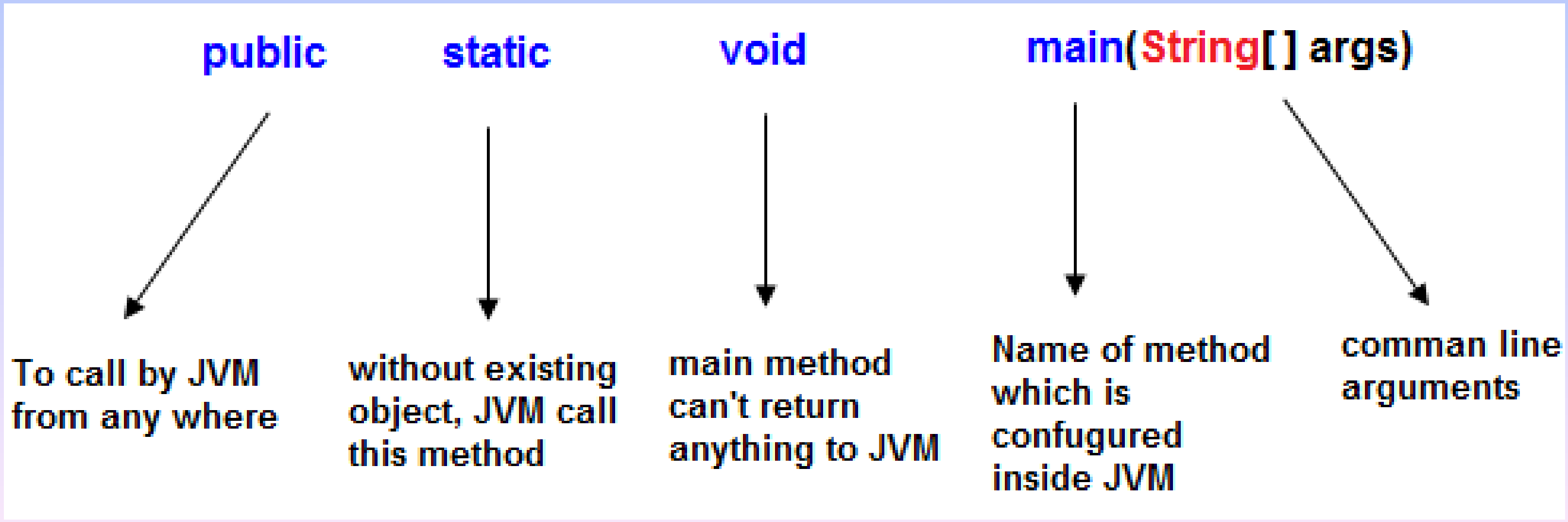
```
package details               →   import java.io.*
class className               →   class Sum
{
Data members;                 →   int a, b, c;
user_defined method;          →   void display();
public static void main(String args[])
{
Block of Statements;          →   System.out.println("Hello Java !");
}
}
```

Main() Method in Java

- **main()** method is starting execution block of a java program or any java program start their execution from main method. If any class contain main() method known as main class.
- Syntax of main() method:

Syntax

```
public static void main(String args[])
{
.......
.......
}
```


public
static
void
main(String[] args)
To call by JVM from any where
without existing object, JVM call this method
main method can't return anything to JVM
Name of method which is confugured inside JVM
comman line arguments