INDIAN INSTITUTE OF TECHNOLOGY MADRAS

सिद्धिर्भवति कर्मजा

IIT Madras

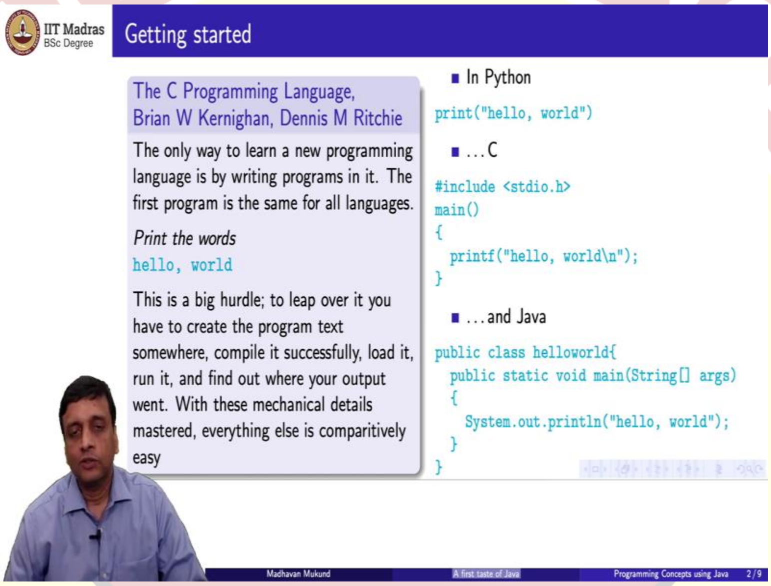
ONLINE DEGREE

Programming Concepts Using Java
Online Degree Programme
B. Sc in Programming and Data Science
Diploma Level

Prof. Madhavan Mukund
Department of Computer Science
Chennai Mathematical Institute

Lecture 7
A First Taste of Java

(Refer Slide Time: 00:18)



Getting started

The C Programming Language, Brian W Kernighan, Dennis M Ritchie

The only way to learn a new programming language is by writing programs in it. The first program is the same for all languages.

Print the words
`hello, world`

This is a big hurdle; to leap over it you have to create the program text somewhere, compile it successfully, load it, run it, and find out where your output went. With these mechanical details mastered, everything else is comparitively easy

- In Python

```
print("hello, world")
```

- ...C

```
#include <stdio.h>
main()
{
  printf("hello, world\n");
}
```

- ...and Java

```
public class helloworld{
  public static void main(String[] args)
  {
    System.out.println("hello, world");
  }
}
```

So, let us get started with Java. So, a starting point is a famous reference which comes from a book by Kernighan and Ritchie which is a very well known book introduced in the C language almost 50 years ago. And in this they suggested that to learn a programming language you really have to get started with it. So, you need to do something and what is better than to get the program to say something.

So, they came up with this iconic hello world as the first message you should be able to print from your programming language. So, they said that the first program you should always be able to write in any language is hello world and if you get over this if you are able to actually somehow edit this program save it compile it and run it you overcome the first major hurdle to learning programming which is to get the mechanics of running a program.

And once you get over this hurdle then everything else is easy. To be that as me this has become a sort of a symbolic challenge to see how easy or difficult it is to program in a given language which is how difficult is it to put out a program that does hello world. So, the language that we have been seeing so, far in this program in this course has been Python and Python is particularly good in this respect.

Because the simplest Python program for hello world is just a single print statement it just prints hello world one line cannot expect to do anything less than that and this is what makes Python for many people a very attractive language to start programming. There is not a big syntactic barrier to starting to write code. In C which is the language which Kernigan and Ritchie were themselves teaching in that book it is a little bit more complicated.

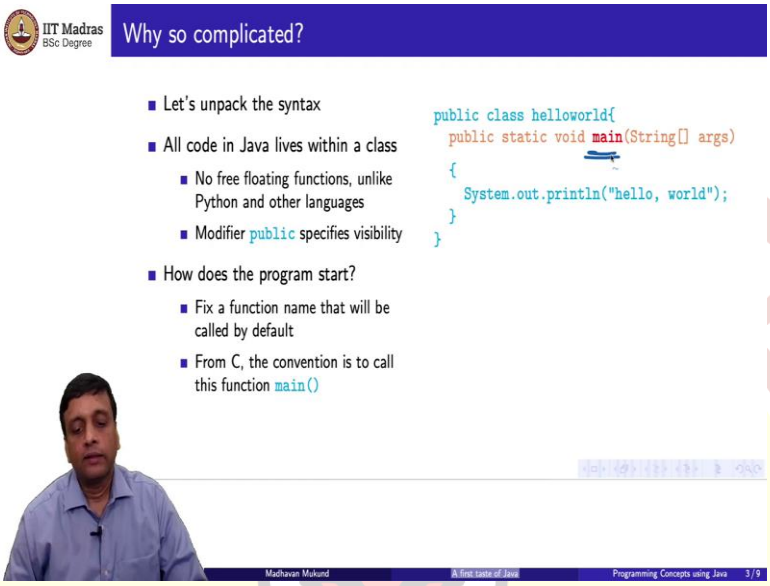
So, one thing is that you need to make this print statement available. So, this is the equivalent of an import statement in Python. So, the C programming language does not have print available by default. So, you have to include the standard input output library. The other thing is that C everything has to live inside a function. So, you have to create a function for this and by convention as we will see the program the function that you start running is always called main.

And finally you have the equivalent of the Python print statement except it is called printf and the other thing is that you have to explicitly put a character called new line to make it print and go to a new line. So, this is already a little bit more complicated in Python and now let us look at the corresponding Java program which does this. So, this is really quite scary to look at. So, buried inside it you see this much which looks familiar.

So, you have a print statement which takes as argument the string hello world. So, presumably that is the job that we are doing. Now like C it is enclosed inside a function definition. Unlike C which just has main it has a lot of stuff around this function definition and certainly unlike C it has a further layer of complexity because we actually have put it inside a class.

So, why should we need a class to print a string. So, let us try to get into this complex definition and understand little bit about why all this machinery is there and though it is of course not going to be possible to explain all of Java with respect to this toy example. It will give us a good idea of the kind of things that we will encounter as we go ahead.

(Refer Slide Time: 03:42)



IIT Madras
BSc Degree

Why so complicated?

- Let's unpack the syntax
- All code in Java lives within a class
  - No free floating functions, unlike Python and other languages
  - Modifier public specifies visibility
- How does the program start?
  - Fix a function name that will be called by default
  - From C, the convention is to call this function main()

```
public class helloworld{
  public static void main(String[] args)
  {
    System.out.println("hello, world");
  }
}
```

Madhavan Mukund　A first taste of Java　Programming Concepts using Java　3 / 9

So, the first thing that we have to do when we unpack the syntax is understand why there is a class out there. So, why is this class definition here. So, remember that Java is designed bottom up as an object oriented programming language it came into being with the philosophy that it would be object oriented. In an object-oriented language as we remember we define functions in the context of objects.

We do not write functions to which we pass data we create data on which we can apply functions we send messages to objects we tell an array to sort itself for example. So, therefore every function in a truly object oriented language has to be sitting inside a class it has to be defined only in the context of an object. We cannot have free floating function sitting out there because we are not supposed to be passing things to functions we are supposed to be invoking functions on objects.

So, every function must live in a class. So, must this one and therefore we have to create a class the other part of it is that we have to know whether this class is available or not. So, remember that we said that in the Python implementation of objects there is no distinction between public and private. So, even though you have encapsulation and things are kind of modularly written there is nothing to prevent you from going inside and adding a field or modifying a value and so on.

Now Java takes object oriented notions very seriously. So, it tries to separate the public and the private quite clearly and this has to be done by declaring what is public and what is private. So, clearly we need to execute this function inside the class. So, the class itself is not public we cannot execute this function. So, we also have in addition to the class definition we have this extra modifier as we could call it in type in Java saying that this class not only is there.

But it so it exists only for the sake of defining a function because we cannot define a function without a class but this class must be public. So, now we move inside and the next thing we encounter is the function itself. So, like in C we need to create a function now why do we need to create a function? So, remember that this is not a language which we are going to interact with.

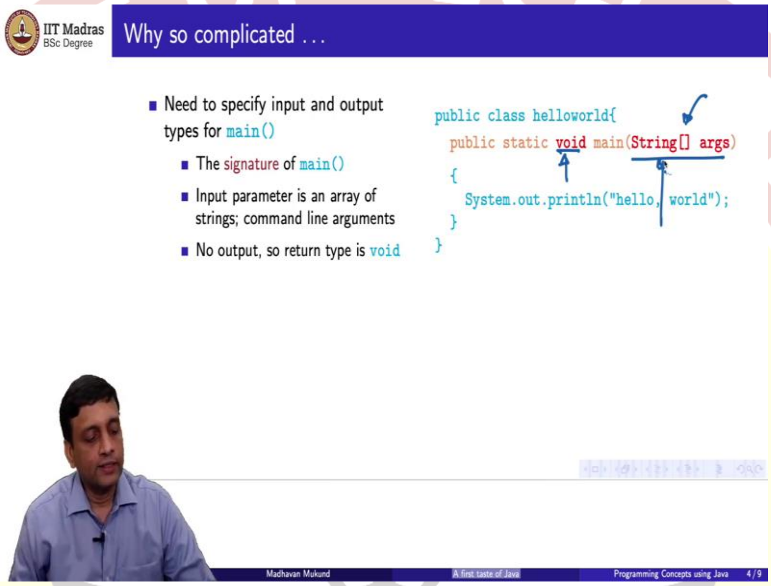
So, when we are writing Python code very often we are interacting with it through something like a Jupiter notebook and if we are not interacting with we call Python as a standalone program then it just goes down because not all Python code has to sit inside a function. So, you can just write Python programs which are a sequence of statements. So, Python will just take a piece of Python code written in a file and execute statement by statement top to bottom if they are function definitions it will understand the definitions if they are not definitions you will try to execute them.

So, there is no problem with the Python program we just give it is a sequence of instructions. Instructions may be function definitions there may be actual statements you just execute them top to bottom. But if you have a language like C or Java where everything must be inside a function then you cannot just start executing but you do not know where to execute.

So, where do you start because everything inside a function which function starts. So, there has to be a convention and. So, C actually began this convention that the function which begins everything the thing that kicks off the whole thing is the one that is called main. So, there is one special function called main and all execution starts there. So, that is why when we defined this hello world in C we have to create a function called main.

So, Java inherits the same convention so to speak. So, in Java all execution starts in main. So, that is the reason we have a function called main. So, we have to write the function inside a class but we have to have a designated function to start with and that is main.

(Refer Slide Time: 07:24)



IIT Madras
BSc Degree

Why so complicated ...

- Need to specify input and output types for main()
  - The signature of main()
  - Input parameter is an array of strings; command line arguments
  - No output, so return type is void

```
public class helloworld{
  public static void main(String[] args)
  {
    System.out.println("hello, world");
  }
}
```

Madhavan Mukund A first taste of Java Programming Concepts using Java 4 / 9

Now the next part is what is around the main. So, what is inside the brackets and what is just before main. So, this is the so called signature of me. So, with every function it takes a certain kind of arguments and it returns a certain kind of return value now. You might think that a function like name which is called from nowhere because it starts off on its own spontaneously as it were nobody is there to pass it any arguments.

So, why should it take any arguments and similarly it is certainly not going to return anything to anything that called it because it is not being called by any function. So, there is no return value. So, the fact that there is no return value is clear so the return value is the one that is outside the function. So, Java is not optional you always have to provide the return value.

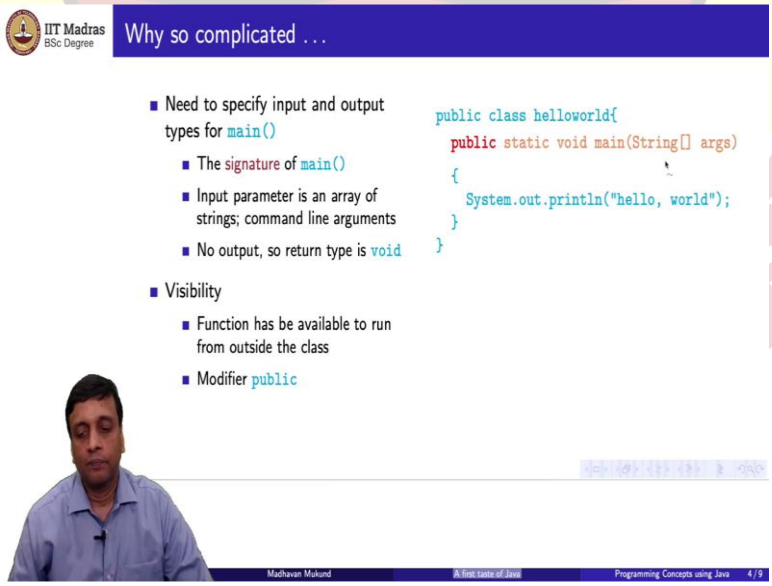
So, if your return value is nonsensical or it does not exist or whatever then you use the special type called void. So void means empty so, void is a type; which contains no values. So, void is essentially the type equivalent of a Python word like none for a value which does not exist. So, here it is saying it is a type a type is a collection of possible values and operations on them.

So, void is an empty type it has no values. So, the value returned by main is of this type. So, therefore it will never return anything that is what it means. This part is a little bit more complicated and we may or may not look at it in more detail as we go along but though you cannot of course call for main with any arguments because you are not calling it from inside another piece of Java code.

When you call it from the command line you can pass arguments through what are called the command line arguments you can write the Java program name followed by certain other things and these other things will be implicitly passed as arguments domain. So, this is now if you as we go ahead we will see that this stands for an array of strings. So, we write the name of the program and then we write a sequence of words after it each of them is passed in turn as an argument and that is all.

So, this void and this string array of called args are the signature of this function that is the input and output type.

(Refer Slide Time: 09:26)



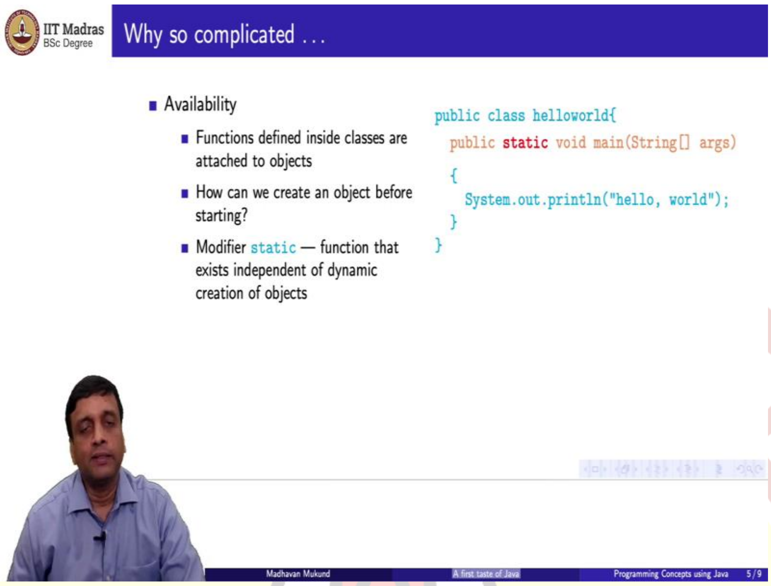
Why so complicated . . .

- Need to specify input and output types for main()
  - The signature of main()
  - Input parameter is an array of strings; command line arguments
  - No output, so return type is void
- Visibility
  - Function has be available to run from outside the class
  - Modifier public

```
public class helloworld{
  public static void main(String[] args)
  {
    System.out.println("hello, world");
  }
}
```

Now just like the class has to be public or private the function has to be public or private and obviously we want to execute this even though we are not executing it directly somebody has to execute it. So, it has to be executed from outside the class and therefore it must be public that is why we have to announce that this function is public. Now what about the static what is static.

(Refer Slide Time: 09:48)



Why so complicated . . .

- Availability
  - Functions defined inside classes are attached to objects
  - How can we create an object before starting?
  - Modifier static — function that exists independent of dynamic creation of objects

```
public class helloworld{
  public static void main(String[] args)
  {
    System.out.println("hello, world");
  }
}
```

So, we are in a peculiar kind of catch 22 of our own making. So, remember that Java has decided because it is object oriented that every function must live inside a class. Now how do we invoke functions on class of where we have to create objects. So, we remember that in Python we created this class called point and we defined a function called translate which would take x y and move it to x + dx , y + dy.

Now obviously we cannot call translate until we have a point because there is no function translate on its own it only exists in the context of one point or another point. So, we have to create an object in order to invoke the function. Now here we have a function because we have put it inside a class called hello world. So, we have to create some object of type hello world in order to invoke main.

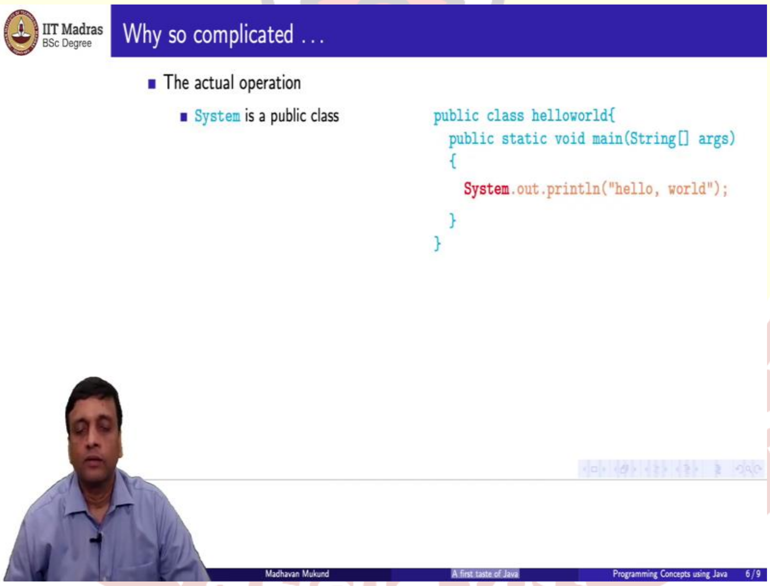
Who is going to create that object it has to be some code but which code is that nothing is running. So, we have this cache 22 as it is called we have this circular thing where we need some code to pick off main and we need main to kick off the code. So, clearly you have to break this cycle somehow. So, that is broken by this modifier called static. So, static says that this definition that we have here inside a class is actually not tied to any object of that class.

So, here it is a function. So, it says that this function is you can think of it, it is more or less like a normal function. So, this is like a function that you would find in Python outside with a depth statement which exists independent of whether you have created objects or not except that here it is defined inside a class and then kind of artificially made into this category by giving it a static modifier.

So, static says this function can be invoked without having to create an object of that class. Many other functions that we can think of will behave like that for instance here we have one already which we will come to when we do it the print statement you do not want to create an object to print but even the familiar math functions if you want to do sine, cosine, square root and they sit inside some library.

Now in Java that library would be a kind of a class and do we want to create a math object before we can use that no we would like to available. So, this idea of having static objects, static functions, static values will be important to us.

(Refer Slide Time: 11:57)



IIT Madras BSc Degree

Why so complicated . . .

- The actual operation
  - System is a public class

```
public class helloworld{
  public static void main(String[] args)
  {
    System.out.println("hello, world");
  }
}
```

Madhavan Mukund · A first taste of Java · Programming Concepts using Java · 6 / 9

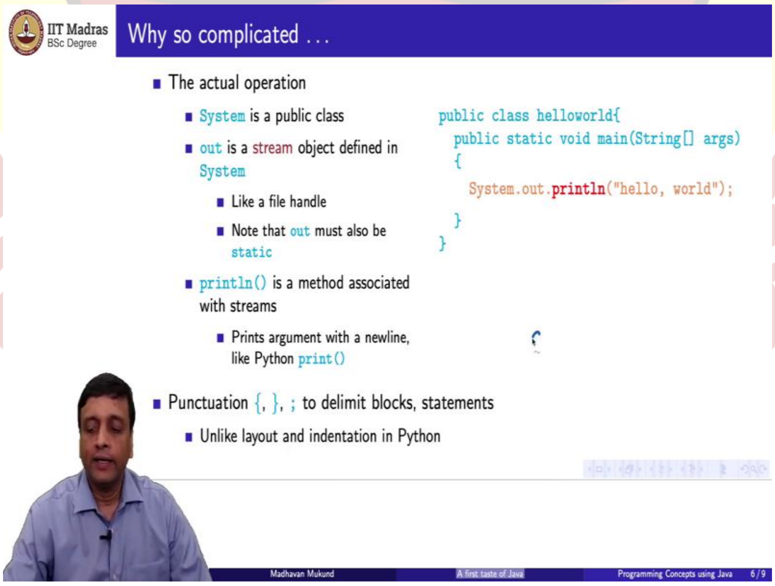
So, that takes care of the two outer lines of code in this program. So, now we come to the actual function. So, here we can see that deep inside there is something which looks similar to what we saw in Python and what we saw in C which is a print statement which takes an argument and prints it. But remember in Java this print statement must live in some class it cannot just be on its own.

So, that class is called system. So, that is why we have to put it with respect to system. So, the syntax here is similar to the Python thing where we use libraries. So, remember Python will say things like math dot something math dot square root. So, you would say that this is the function square root which is inside the library math. So, here instead of a library is a class. So, this is the function println inside the class system.

So, that is what the system is telling us. So, system is a public class which contains this print state print function but the print function is not again a function which can be applied in general it has to print somewhere. So, in Python for instance when you do file input and output then what you have to do is you have to associate what is called a file handle. So you connect your program to the file through kind of a pipe and you write or read from that pipe and then the system outside has to do the job of converting what is in the pipe into something on the disk and vice versa.

So, you do not directly read from the disk or to the disk. So, you can do this in abstract way by dealing with these file handles and if this file handle is not really going to a file but it is going to a kind of from coming from the keyboard or going to the screen or something then this is called a stream.

(Refer Slide Time: 13:42)



IIT Madras BSc Degree

Why so complicated . . .

- The actual operation
  - System is a public class
  - out is a stream object defined in System
    - Like a file handle
    - Note that out must also be static
  - println() is a method associated with streams
    - Prints argument with a newline, like Python print()
- Punctuation {, }, ; to delimit blocks, statements
  - Unlike layout and indentation in Python

```
public class helloworld{
  public static void main(String[] args)
  {
    System.out.println("hello, world");
  }
}
```

Madhavan Mukund | A first taste of Java | Programming Concepts using Java 6 / 9

So, out is a stream it is a stream of data which is going out somewhere. So, the Java program just sees as a pipe where it will write out data and what this is saying is that we are printing with respect to the particular pipe called out. So, out is a specific stream which will feed in our case typically to the screen on which we are viewing this Java program. So, remember that this out is an object in some sense and it exists because we have not created anything clearly in this code there is no line which created out.

So, out must already exist just like main already existed. So, out is also some kind of a static object. So, out is a stream object which is static and defined inside the public class system and we are invoking println with respect to this particular stream object. So, finally we come to println well it is just a print statement. Now in Python print says print with a new line and if you do not want a new line you have to pass an extra argument.

In C printf does not put a new line unless you pass an extra character if you go back and look at that code we saw before there was this "\n" that we added at the end of the string to print a new line. So, Java it turns out has two different functions it has a println and a print. The print will just print what it is given to it and stay on the same line print l and go to a new line.

So, otherwise there is nothing very complicated. So, at its core this is the same function it just is wrapped up in all the syntax. Now there is of course with respect to Python one more level of difficulty in syntax which we have not seen before which is all this punctuation. So, in Python we write def something and then we put a colon and then we indent our code or we write for something put a colon and then we indent our code.

So, there is implicitly a notion of a block of code which is tied to that form or a block of code which is tied to the definition. So, how do you demarcate the beginning and end of this block in Python this is done by layout. So, you indent and everything which is at the same level of indentation is part of the block and once you go out of that level and come out to a higher level of indentation the block is over.

So, you have to be very careful about how you lay it out but the advantage is you do not need any other kind of markers to indicate the beginning and the end. Now in most programming languages this is done syntactically by using markers and the normal way of marking it is to put an opening and a closing bracket or a brace especially. So, this is the opening brace it says that this is the block of code defining the class begins here and the matching one is here.

And then inside that there is a function the beginning of the block defining the function is here and the end is here. So, you use explicit begin and end brace. So, these have to be matched correctly like all brackets do in any expression that you write in arithmetic also you would have to match the brackets properly on intersect brackets. But you have to match them and for every opening brace there will be a corresponding closing brace and that denotes a block of code.

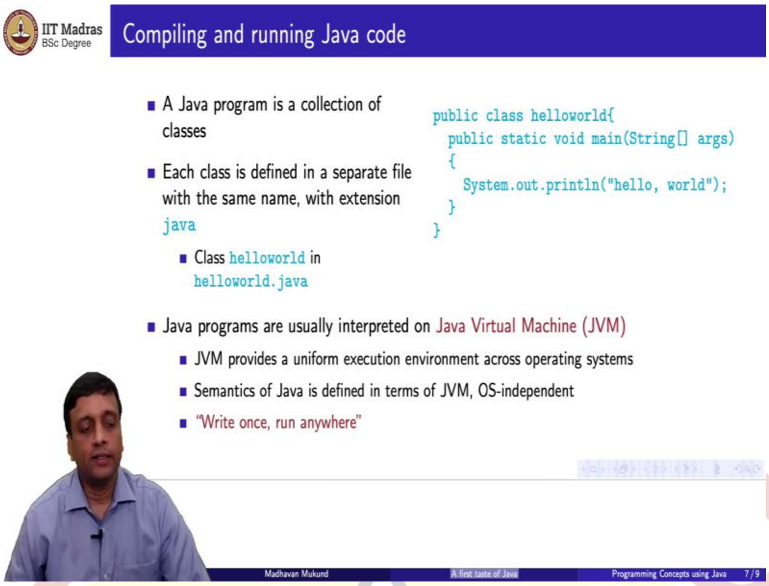
So, the block of code could denote in this case the body of a function the body of a class it could donate like for example the scope of a loop or the scope of an if or the scope of an else as we will see. So, this is compulsory and the other thing which is compulsory in Python in Java is that every statement must end with a semicolon. So, this means of course you can write all of your code in one line because now the layout is not important.

But then of course it will become unreadable but in principle you can write a Java program as you wish. The new line has no real significance in laying out a Java program unlike in a Python program nor does the implementation. So, the choice of indentation and where to put new lines is more a question of style as far as the programmer goes for readability but it is not required for understanding the code from the compiler whereas in Python the layout is crucial to understanding the code.

So, there are tradeoffs is much easier to read Python code if you do not have all these braces but also it is a little tricky because sometimes by if you accidentally get the indentation wrong or if in the process of transferring the file from one person to another person and back the spaces get changed little bit then unintentionally you might end up changing the meaning of the program and not realizing it until later. So, it is a source of bugs also.

So, it is a question of whether you like it or not but almost every programming language other than Python will actually use braces and semicolons to delimit blocks and statements. So, that deconstructs this program. So, we have this program what do we do with it.

(Refer Slide Time: 18:10)



IIT Madras
BSc Degree

Compiling and running Java code

- A Java program is a collection of classes
- Each class is defined in a separate file with the same name, with extension java
  - Class helloworld in helloworld.java

```
public class helloworld{
  public static void main(String[] args)
  {
    System.out.println("hello, world");
  }
}
```

- Java programs are usually interpreted on Java Virtual Machine (JVM)
  - JVM provides a uniform execution environment across operating systems
  - Semantics of Java is defined in terms of JVM, OS-independent
  - "Write once, run anywhere"

Madhavan Mukund A first taste of Java Programming Concepts using Java 7 / 9

So, in general as we said everything in Java has to be written inside a class. So, therefore a Java program is a collection of classes. So, how does Java keep track of this well what Java says is that each class must be in a separate file and the name of the file must match the name of the class. So, here we have chosen the name you can choose any name you want. So, we have chosen the name hello world for this class and that means we must put it in our file called hello world with an extension dot Java.

So, the extension dot java is compulsory tells us a Java program but what we write before the dot is exactly the name of the class the class name inside in the class and the name of the file should match otherwise Java will get confused. So, we put the class hello world in hello world.java. Now eventually this will run but where will it run. So, when we use languages like C or C ++ of course we compile them.

So, that is why we said that when we compile them and we run them we are not interacting with them. So, we have to use some convention that they start with main and so on. But the important thing is that they are running on the hardware that you are using. So, if you are running it on say a windows machine on an Intel CPU or if you are running it on a MAC on an M1 chip or if you are running it on a Linux machine on an amd chip the underlying architecture and the machine language is different.

So, you start with the same high level language but your compiler will produce code for the given architecture. Now usually these architectures are similar but sometimes there are some subtle differences. So, code that runs properly on one machine may or may not behave the same on another machine and this used to be even more problematic 15 20 years or 25 years ago when Java came into being.

So, Java is roughly from the mid 1990s. So, it was created in a scenario where there were frequent bugs resulting from the fact that code which was tested and verified on one system would not run in a different operating environment. So, the solution that Java came up with is to produce a uniform environment for running Java code. So, they said let us not think of Java code as running on this machine or that machine but it is running on an abstract machine which we call a Java virtual machine.

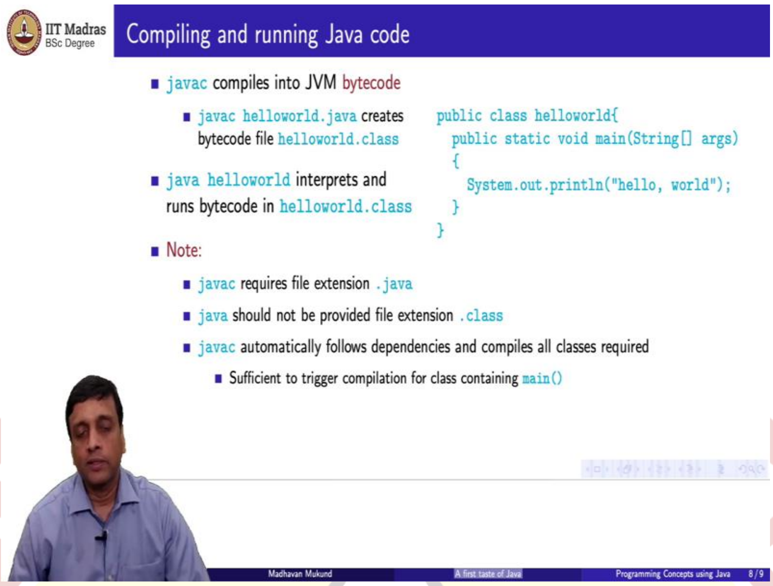
So, Java code is designed to run on an abstract virtual machine and this abstract virtual machine is then implemented with respect to different architectures. So, the problem of implementing on the architecture is restricted to the virtual machine. You have to make sure that the virtual machine instructions are converted to the local machine instructions once and for all. Now each individual Java program is written in terms of the virtual machine.

So, then that part of it will work on any implementation because all the virtual machines are supposed to behave the same whether the underlying architecture is running on one operating system or another one chip or another. So, this JVM provides a uniform execution environment and it also allows us to give a uniform semantics for Java. If somebody says what is the maximum size of an integer in Java you do not have to say oh if it is a 32-bit architecture business and if it is a 64-bit architecture it is that and so on because the JVM will decide.

If the JVM is 64-bit JVM it is so, much if it is 32-bit available. So, much it does not matter whether it is running on this chip or that. So, it is OS independent. So, in fact one of the big slogans when Java was introduced was the ones run anyway and this was particularly important because it also coincided with the time when the internet was getting more popular and people were writing code which would run on different machines and transporting it across and so on.

So, it is also a need for running code in different environments. So, it was born out of a necessity and it is a very useful thing to have in general that Java code is portable.

(Refer Slide Time: 21:52)



Compiling and running Java code

- javac compiles into JVM bytecode
  - javac helloworld.java creates bytecode file helloworld.class
- java helloworld interprets and runs bytecode in helloworld.class
- Note:
  - javac requires file extension .java
  - java should not be provided file extension .class
  - javac automatically follows dependencies and compiles all classes required
    - Sufficient to trigger compilation for class containing main()

```
public class helloworld{
  public static void main(String[] args)
  {
    System.out.println("hello, world");
  }
}
```

So, we still have a machine it is not the machine on which we are actually working but there is a layer on top this JVM. So, our Java code has to be translated into that machine. So, we still need a compiler. So, this is called javac or java c. So, Java compiles your Java file into what is called bytecode. So, this is very similar to any compiler but instead of compiling it for a specific machine architecture it is compiling it to this hypothetical virtual machine code.

So, this is called Java bytecode and this will sit in a file called helloworld.class. So, now this dot class, so, we started with helloworld.java and javac will produce as output bytecode which it will put in the same class name but with an exciting dot class. Now this is a little bit confusing but this class has nothing to do with the object oriented class it is just it could have chosen any other name could be called cake or it could be called butterfly or whatever but they happen to choose the word class.

So, your Java will take something in a file.java and you produce it in the same file.class. So, this is now the code that will run but you have to run it and how do you run it well you have to invoke an interpreter. So, the interpreter is called Java. So, Java running on this class file will run it on the JVM. So, you need a JVM installed on your machine and there are JVM installations available for almost anything you have to search and download and install Java development environment as is called a JDK and it you can run.

So, Java hello world will run this bytecode and the promise is that this will run the same on your JVM as it will run in any other JVM. So, you can take in principle this .class file and run it anywhere you do not have to take the source code and recompile it on another machine. So, there are some minor irritations with this whole thing. So, one thing is the way that you call.

So, when you use the compiler you must of course call it on a file which is has a .java extension but you must pass the extension as part of the command. So, you must say javac helloworld.java you cannot just say javac helloworld and expect it to final over I do not know why it should be easy but it is part of the design. Now it will produce this .class file but Java the interpreter will not allow you to say .class you have to say Java hello world and then it will look for .class.

So, this minor, so, for the compiler you provide the extension when you call it for the interpreter you do not provide these calls. The other potential problem is this business of having different classes for different files for different classes. So, remember that in Java everything must sit in a class and every class must sit in a file of that name. So, you cannot put 20 classes in one file because the class name must match the file name. So, every class is a separate file.

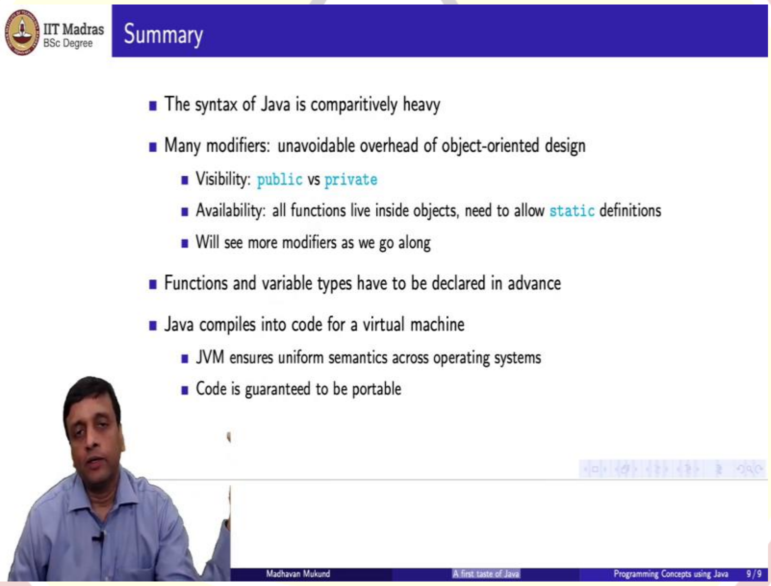
So, if this were required that we must compile every class separately then it will be very tedious if you are using 10 or 20 different data types or various objects in your types of objects in your code then each time you compile your visit and meticulously compile unfortunately that is not required. Remember that this main is the starting point. So, among all your classes there should be one which defines a name. So, that is where things are going to start.

So, javac the compiler is invoked on that class. Now inside that class you might define objects of other classes. So, this is where Java uses the fact that the name of the class and the name of the file match. So, if you suddenly say that I have defined a point inside this particular function main then it will go and look for a point the Java because the class point must be in a file point.

Now if point dot Java in turn uses some other file then it will go a class it will go and look for that file. So, by following the chain of classes now how do these classes come about remember everything is declared. So, you have to declare that I have a variable of type point I have a variable of type colour I have a variable of like that. So, for every declaration that Java finds in your code it knows that that corresponding class file must also be compiled.

So, Java compiler is smart enough to follow this chain of dependencies and compile everything that is needed and if something is missing it will tell you. So, fortunately though you have to keep track of these different files one per class you do not have to manually compile them they get compared automatically.

(Refer Slide Time: 26:21)



IIT Madras
BSc Degree

Summary

- The syntax of Java is comparitively heavy
- Many modifiers: unavoidable overhead of object-oriented design
  - Visibility: `public` vs `private`
  - Availability: all functions live inside objects, need to allow `static` definitions
  - Will see more modifiers as we go along
- Functions and variable types have to be declared in advance
- Java compiles into code for a virtual machine
  - JVM ensures uniform semantics across operating systems
  - Code is guaranteed to be portable

Madhavan Mukund | A first taste of Java | Programming Concepts using Java 9 / 9

So, what we have seen is that compared to Python certainly and even compared to what little we saw of that example in C. The syntax of Java is quite heavy it is a lot of baggage which seems to be there in the way of writing even simple code but unfortunately this is a byproduct of the design choices that have gone into the language itself. So, it is a purely object oriented language and it wants to clearly maintain this public interface versus the private implementation idea.

It wants to make sure that programming is disciplined without the programming without having to rely on rely on the programmers discipline. So, you want to make sure the compiler can enforce this discipline. So, you need these modifiers public and private you need to create objects. You need to have functions available before you create objects. So, you need these modifiers like static and so on.

The other thing is that like languages like C and C++ you need to declare functions and variables in advance. So, this helps in two ways one is as a programmer it helps you from making mistakes and for the compiler it makes it easy to keep track of storage and what kind of values to use and so on. And finally the selling point of Java is these once run anywhere. So, Java is designed to be a portable language.

So, it is defined in terms of this virtual machine. So, Java compiler creates this bytecode for this virtual machine and this bytecode is guaranteed to run in the same way across architectures. So, you have portable code.