GILBERT KOFI
GRUNITZKY

PROGRAMMING

JAVA

Course Outline

1. Introduction to OO programming
2. Language Basics – Syntax and Semantics
3. Algorithms, stepwise refinements.
4. Quiz/Assignment (
5. Repetitions (for loops)
6. Writing simple classes
7. Arrays, methods and parameters
8. Quiz 2/Assignment
9. Inheritance
10. File input/output

Introduction to java

Characteristics of java

1. Simple
2. Familiar
3. Object Oriented
4. Portable
5. Robust and Secure - Java provides extensive compile time and run-time checking.

Types of java Applications

1. Java Console Application – Can only display textual data
2. Java GUI Application – Development of graphical user interface
3. Java Applets – Embed a program directly into a web browser.

Console Application

• Usage

– Limited user interface

– Applications that run in the background

– Quick test applications

• Limitations

– No mouse support

– No graphical/windows support

Graphical User Interface

• Usage

– Working directly with the user

– Applications that must display graphical data

• Limitations

– More complex to setup than console application

– Less convenient to run in the background

Java Applet

• Usage

– When your application have to run with a browser

– Applications that display animation

• Limitations

– Cannot save data to the user's disk

– Cannot make use of most emerging technology

Different forms of Java

• Java Server Page (JSP) –helps to create dynamic web pages.

• Java 2 Enterprise Edition (J2EE) – used to transfer XML structured documents.

• JavaBeans – use to create new and advance applications (reusable software component)

• Java Mobile – used to develop mobile applications

• Java DB – used to create distributed applications

Why java?

• Lots of applications and websites won't work unless you have Java installed.

• From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

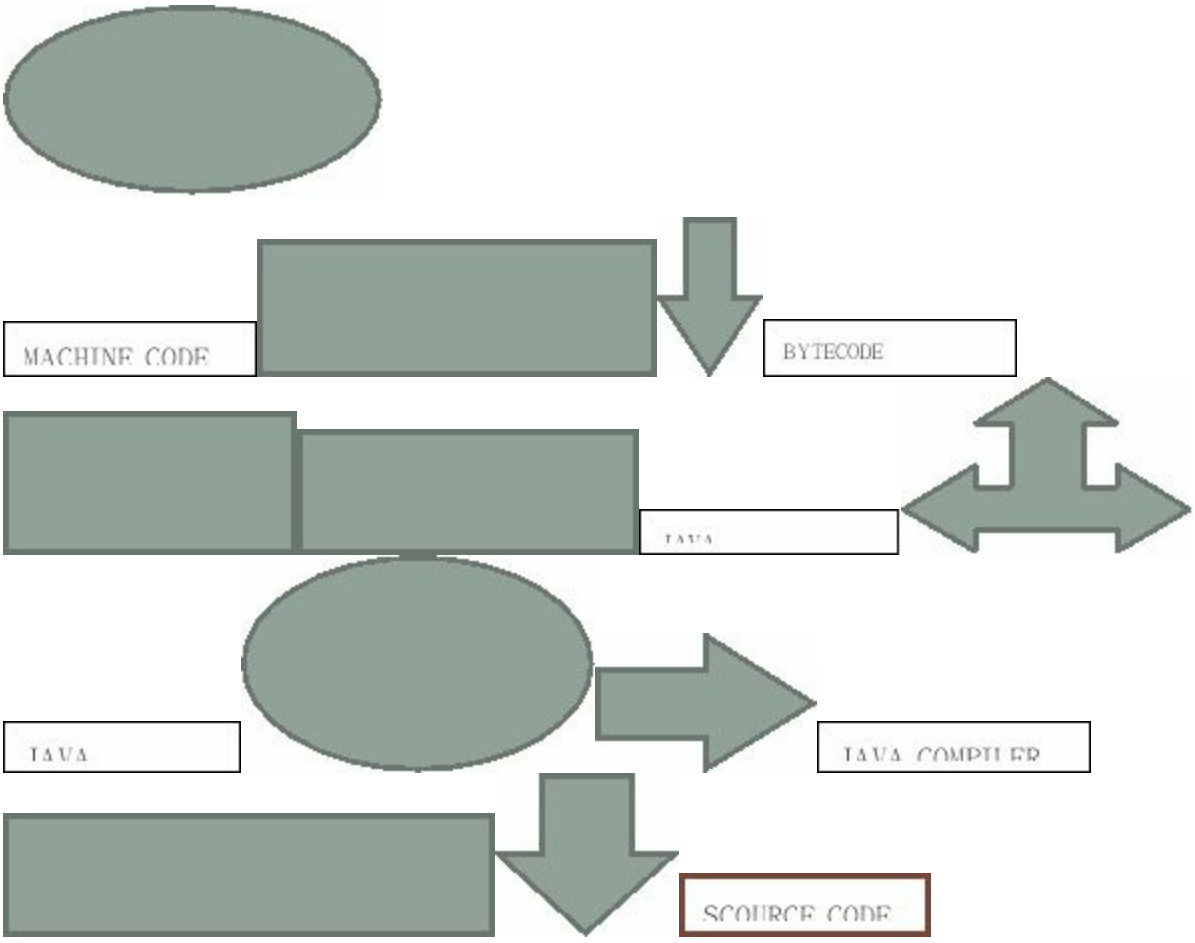
• Most of the popular applications for phones were developed using java programming language. E.g. Opera Mini, Skype Lite, eBuddy IM Service, Gmail for Mobile, Google Maps Mobile etc

Creating a java program

• What is the difference between a compiler and an interpreter?

• A Java program is both compiled and interpreted.

• Compilation happens just once; interpretation occurs each time the program is executed


MACHINE CODE
BYTECODE
JAVA
JAVA
JAVA COMPILER
SOURCE CODE

Running a Java Program

• Requirements to run a program

– Java Development Kit (JDK)

– Text editor

• Integrated environment

– Netbeans IDE

– Eclipse

First java Program

```
1     // Our first example
2       // The famous hello world !!!
3       public class Hello {
4       // main method begins execution of Java application
5        public static void main( String args[] )
6     {
7                     System.out.println( "Welcome to Java
Programming!" );
8       }
9    // end method main
10  }
11   // end class Welcome1
```

• Output

Welcome to Java Programming

Comment

1 // Our first example

– Comments start with: //

• Comments ignored during program execution

• Document and describe code

• Provides code readability

– Multiple line comments: /* ... */ /* This is a multiple line comment. It can be split over many lines */

2 // The famous hello world !!!.

– Another line of comments

– Note: line numbers not part of program, added for reference

Class definition

3

– Blank line • Makes program more readable

• Blank lines, spaces, and tabs are white-space characters

– Ignored by compiler

• Naming classes: capitalize every word

– SampleClassName

Class Definition

– Name of class called identifier

• Series of characters consisting of letters, digits, underscores (_) and dollar signs ($)

• Does not begin with a digit, has no spaces

• Examples: Welcome1, $value, _value, button7 – 7button is invalid

• Java is case sensitive (capitalization matters)

– a1 and A1 are different

– Use of public keyword

• Certain details not important now

• Mimic certain features, discussions later

Parts of program

3 public class Hello {

– Saving files

• File name must be class name with .java extension

• Hello.java

– Left brace {

• Begins body of every class

• Right brace ends definition (line 13)

7 public static void main(String args[])

– Part of every Java application

• Applications begin executing at main

– Parenthesis indicate main is a method (later)

– Java applications contain one or more methods

Parts of program

7 public static void main(String args[])

• Exactly one method must be called main

– Methods can perform tasks and return information

•void means main returns no information

• For now, mimic main's first line

8 {

– Left brace begins body of method definition

• Ended by right brace } (line 11)

9 System.out.println("Welcome to Java Programming!");

– Instructs computer to perform an action

• Prints string of characters

– String - series characters inside double quotes

• White-spaces in strings are not ignored by compiler

– System.out

• Standard output object

• Print to command window (i.e., MS-DOS prompt)

– Method System.out.println

• Displays line of text

• Argument inside parenthesis

– This line known as a statement

• Statements must end with semicolon ;

12 } // end method main

– Ends method definition

13 } // end class Hello

– Ends class definition

– Can add comments to keep track of ending braces

Displaying multiple line of text

• Newline characters (\n)

– Interpreted as “special characters” by methods System.out.print and System.out.println

– Indicates cursor should be on next line

9 System.out.println("Welcome\nto\nJava\nProgramming”);

– Line breaks at \n

• Usage – Can use in System.out.println or System.out.print to create new lines •System.out.println("Welcome\nto\nJava\nProgramming!");

Console input and output

Programming Errors

• Syntax errors.

• Logical errors.

2

Reading Strings/ numbers in

java

- Libraries in java are called packages
- Package is a collection of classes
- Classes and object?
- Two classes are available:
  - BufferedReader Class
  - Scanner Class

Using Scanner Class

- Write a program to accept a string from the keyboard and echo it on the screen?

Reading integers value in java

- Write a program to accept the age of a user and echo it on the screen
- Modify the program to display name and age of the user.

Example programs

- Write a program to accept two whole numbers and display the sum on the screen.
- Write a program to display the circumference of a circle.
- Write a program to compute the volume of a sphere.

Assignment

- Write a program to compute the roots of a quadratic equation.

$$X = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Question

•What is the value of x?

double x = 1/2.0 + 1.0/2 + 1.0/2.0 + 1/2;

Questions

• What will be the values stored in the variables x and y respectively after the following 4 statements have been executed? int x = 5; int y = 10; x = y; y = x;

Formatting numeric variables

• The use of printf

• Date format

Primitive Data Type

Type name	Range
boolean	True or False
char	Single character
byte (8 bits)	-128 to 127
short (16 bits)	-32,768 to 32,767
int (32 bits)	-2,147,483,648 to -2,147,483,647
long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float (32 bits)	-3.4E+38 to 3.4E+38
double (64 bits)	-1.7E+308 to +1.7E+308

Arithmetic

• Arithmetic calculations used in most programs

– Usage

• * for multiplication

• / for division

• +, -

• No operator for exponentiation

– Integer division truncates remainder 7 / 5 evaluates to 1

– Modulus operator % returns the remainder 7 % 5 evaluates to 2

Arithmetic

• Operator precedence

– Some arithmetic operators act before others (i.e., multiplication before addition)

• Use parenthesis when needed

– Example: Find the average of three variables a, b and c

• Do not use: a + b + c / 3

• Use: (a + b + c) / 3

– Follows PEMDAS

• Parentheses, Exponents, Multiplication, Division, Addition, Subtraction

Operator Precedence

• Evaluate the following expression

1 + 2 – 3 * 4 / 5

Arithmetic

Operator(s)	Operation(s)	Order of evaluation (precedence)
( )	Parentheses	Evaluated first. If the parentheses are nested, the expression in the innermost pair is evaluated first. If there are several pairs of parentheses on the same level (i.e., not nested), they are evaluated left to right.
*, / and %	Multiplication Division Modulus	Evaluated second. If there are several of this type of operator, they are evaluated from left to right.
+ or -	Addition Subtraction	Evaluated last. If there are several of this type of operator, they are evaluated from left to right.

Equality and relational operators

Standard algebraic equality or relational operator	Java equality or relational operator	Example of Java condition	Meaning of Java condition
Equality			

operators			
=	==	x == y	x is equal to y
	!=	x != y	x is not equal to y
Relational operators			
>	>	x > y	x is greater than y
<	>	x < y	x is less than y
	>=	x >= y	x is greater than or equal to y
	<=	x <= y	x is less than or equal to y

Logical Operators

- and (returns true if both operands are true) - &&
- or (returns true if at least one operand is true) - ||
- not (returns true if operands is false) - !
- exclusive or (return true if only one operand is true) - ^

Pre and Post Increment or Decrement

- Post (a++) or (a--)
- Pre (++a) or (--a)

Example

int n = 2; int value = 2 * (n++); System.out.print (value);

int n = 2; int value = 2 * (++n); System.out.print (value);

Algorithms and Pseudocode

• Algorithm

– A set of steps to solve problem or perform a task.

– A process or set of rules to be followed in calculations or other problem-solving operations, esp. by a computer.

• Pseudocode

– Informal language for developing algorithms

– Helps developer "think out" algorithms

if…else statement

```
if (expression)    statement;
 else    statement;
if (expression) {
        statement;     statement;
        } else {       statement }
```

Question

• Write a program to check whether a person is eligible to vote or not. Minimum allowed age for voting is 18 years.

Question

• Write a program to find largest number between 2 numbers

Question

• Design a program to determine the gross pay for an hourly employee. The employer pays an overtime rate of one-and-half times the regular rate for all hours after the first 40 hours worked. The rate is 12.75 per hour.

Multiple if else statement

• Write a program to compute the grading system for my school.

• Grading scheme

A = 80

B = 70

C = 60

D = 50

E <=49

Switch Statement

• Alternative to if…else statement

Repetition / Iteration

While statement

• Write a program to print from 1 to 10 using a while statement.

• Write a program to print from 10 to 1 using the while statement.

• Write a program to display the squares of the integers between 1 and 10.

do…while statement

int i = 1; do { System.out.println (i); i++; } while (i <= 10);

do…while statement

• Write a program to sum integers read from input (keyboard) until a zero (0) is read.

for Statement

for (initial; test; increment)

for (int i = 1; i <= 5; i++) { System.out.println(i); }

for Statement

- Write a program to display the first five even numbers?
- Write a program to display the first five odd numbers?

questions

- Write a program to print the sum of the first 10 positive numbers.

int sum = 0; For (int i = 1; i <= 10; i ++) { sum = sum + i; } Print sum ;

Question

- Write a program to compute the factorial of any given number.

Question

What is the output of the program?

for (int j = 1; j <= 5; j++){ System.out.println (“*”); }

1. for (int i = 1; i <= 2; i++){
2. for (int j = 1; j <= 2; j++){
3. System.out.print (“*”);
4. }
5. System.out.println ( );
6. }

Example

for (int i = 1; i <= 5; i++) { for (int j = 1; j <= i; j++){ System.out.print (“*”); } System.out.println ( ); }

Example

for (int i = 5; i >= 0; i--) { for (int j = 1; j <= i; j++){ System.out.print (“*”); } System.out.println ( ); }

Example

for (int i = 1; i <= 5; i++) { for (int j = 1; j <= i; j++){ System.out.print (j); } System.out.println ( ) }

What is the output of this program

for (int i = 1; i <= 3; i++) { for (int j = 1; j <= 2; j++){ System.out.print (“*”); } }

What is the output of this program?

for (int i = 5; i >= 1; i--) { for (int j = 1; j <= i; j++){ System.out.print (“*”); } System.out.println ( ) }

Question

• Write a program to compute the multiplication table of any given number.

Expected output

Enter a number : 5

5x1 = 5

5x2 = 10

5x3 = 15 . . .

5x12 = 60

Array

• A special kind of object used to store a collection of data of the same type.

• Arrays are fixed length

• Arrays are zero indexed

Declaring an array

• Data type [ ] name = new data type [size of array]

• Data type name [ ] = new data type [size of array]

• int array [ ] = new int [5];

• String arr [ ] = new String [5];

Assigning values to Arrays

• String arr [ ] = {“Apple”, “Mango”, “Orange”}

• int arr [ ] = new int [5];

• arr [0] = 23;

• arr [1] = 24 ….

• arr [3] = arr [2] + arr[1]

Declaring an Array

• double score [ ] = new double [5];

Question

• Write a program to accept a number of integers and display it on the screen.

Question

• Write a program to accept a number of strings and then display in reverse order.

Classes & Objects

• An object is an instance of a class.

• A class is a template or model or blueprint from which instances are created.

• A Java program can consists of one or more classes

• A class is an abstract description of objects

• Here is an example class:

– class Dog { ...description of a dog goes here... }

• Here are some objects of that class

Classes contain data definitions

• Classes describe the data held by each of its objects

• Example: – class Dog { String name; int age; ...rest of the class... }

• Classes also contains methods

Basic Structure of a Class

• Class Declaration • Variable (Attributes)

• Instance Variable

• Constructor

• Method

• Instance Method

• Class Method

Creating a method

• A method is a program module that contains a series of statements that carry out a task.

```
Public class  first{
Public static void main(String[] args){

        System.out.println("this is my first application");
```

}

}

The Method Header

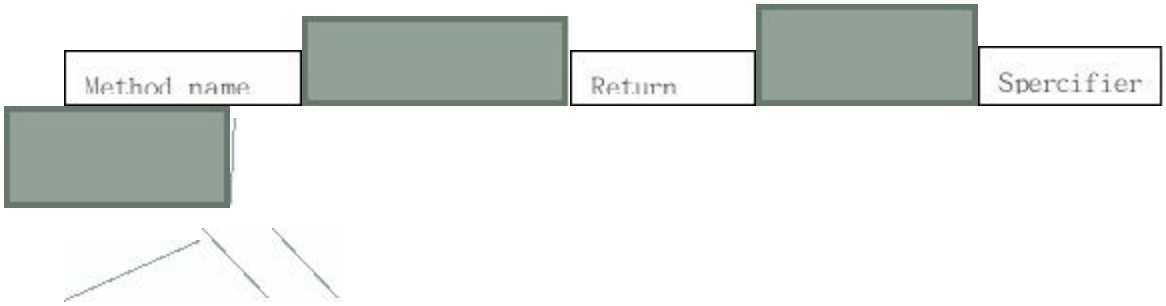
• A Method header consist of the following:

– Access Specifier/modifier

– The return type

– The name of the method

– List of parameters for the methods



Public static void main(String[] args){

Method header

System.out.println(“this is my first application”);

}

Methods with parameters

• A method to compute sum of two integers

Accessor Methods and Mutator

Methods

• A method that accesses an object and returns some information about it, without changing the object is called an accessor method.

• In contrast, a method that modifies the state of an object is called a mutator method

Instance variable

• Attributes or instance variable???

• Each instance must store the values reflecting its current state.

• Declaration of instance variable

Constructor

• The constructor initializes the instance variable of an object.

• The constructor has the same name as the class.

• Usually declared as public and have no return type.

Question

• We want to create a class to model a bank account.