

ENE 212-0083/2017

Paul Moses Mulwa

Object Oriented Programming Assignment 1

ECE 4.2

1. Discuss the Java characteristics.

❑ **Object Oriented:** In java everything is an Object. Java can be easily extended since it is based on the Object model.

❑ **Platform independent:** Unlike many other programming languages including C and C++ when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by virtual Machine (JVM) on whichever platform it is being run.

❑ **Simple:** Java is designed to be easy to learn. If you understand the basic concept of OOP java would be easy to master.

❑ **Secure:** With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

❑ **Architectural- neutral:** Java compiler generates an architecture-neutral object file format which makes the compiled code to be executable on many processors, with the presence Java runtime system.

❑ **Portable:** Being architectural neutral and having no implementation dependent aspects of the specification makes Java portable.

❑ **Robust:** Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

2. How packages differ from a class?

A class is a declaration (and often implementation) which encapsulates the members properties and methods which instances of that type will possess. You can create instances of a class as distinct, individual objects that behave as specified.

A package is simply a namespace under which you can categorize classes, analogous to a "folder" or "path", so you can group classes with related purpose or functionality.

3. When do we use the following symbols? \n \t \r .

☐ \n Newline. Position the screen cursor at the beginning of the next line.

☐ \t Horizontal tab. Move the screen cursor to the next tab stop.

☐ \r Carriage return. Position the screen cursor at the beginning of the current line and do not advance to the next line. Any character written after the carriage return overwrites the character previously output on that line.

4. What is being done in this statement? Scanner input = new Scanner(System.in);

This is a **variable** declaration statement that specifies the type and name of a variable (input) that is used in this program. A **variable** is a location in the computer's memory where values can be stored for use later in the program. All variables must be declared with a name and a type before they are used. A *Scanner* is a built in Java class that enables a program to read data for use in a Page 7 of 55 program and initialize the input variable with equal sign (=). The expression *new Scanner(System.in)* creates a *Scanner* object that reads data typed by the user at the keyboard. *System.in* is a standard input object (Similarly, *System.out* is a standard output object) that enables Java application to read information typed by the user.

5. Explain two kinds of data types available in Java

a. Primitive Data Types

There are eight primitive data types supported by Java. Primitive data types are predefined by the language and named by a key word. Let us now look into detail about the eight primitive data types.

byte:

☐ Byte data type is a 8-bit signed two's complement integer.

☐ Minimum value is -128 (-2^7)

☐ Maximum value is 127 (inclusive)($2^7 - 1$)

☐ Default value is 0

☐ Byte data type is used to save space in large arrays, mainly in place of integers, since a byte is four times smaller than an int.

☐ Example : byte a = 100 , byte b = -50

short:

Short data type is a 16-bit signed two's complement integer.

Minimum value is $-32,768$ (-2^{15})

Maximum value is $32,767$ (inclusive) ($2^{15} - 1$)

Short data type can also be used to save memory as byte data type. A short is 2 times smaller than an int

Default value is 0.

Example : short s = 10000 , short r = -20000

int:

Int data type is a 32-bit signed two's complement integer.

Minimum value is $-2,147,483,648$ (-2^{31})

Maximum value is $2,147,483,647$ (inclusive). ($2^{31} - 1$)

Int is generally used as the default data type for integral values unless there is a concern about memory.

The default value is 0.

Example : int a = 100000, int b = -200000

long:

Long data type is a 64-bit signed two's complement integer.

Minimum value is $-9,223,372,036,854,775,808$ (-2^{63})

Maximum value is $9,223,372,036,854,775,807$ (inclusive). ($2^{63} - 1$)

This type is used when a wider range than int is needed.

float:

Float data type is a single-precision 32-bit IEEE 754 floating point.

Float is mainly used to save memory in large arrays of floating point numbers.

Default value is 0.0f.

Float data type is never used for precise values such as currency.

❏ Example : float f1 = 234.5f

double:

❏ double data type is a double-precision 64-bit IEEE 754 floating point.

❏ This data type is generally used as the default data type for decimal values. generally the default choice.

❏ Double data type should never be used for precise values such as currency.

❏ Default value is 0.0d.

❏ Example : double d1 = 123.4

boolean:

❏ boolean data type represents one bit of information.

❏ There are only two possible values : true and false.

❏ This data type is used for simple flags that track true/false conditions.

❏ Default value is false.

❏ Example : boolean one = true

char:

❏ char data type is a single 16-bit Unicode character.

❏ Minimum value is '\u0000' (or 0).

❏ Maximum value is '\uffff' (or 65,535 inclusive).

❏ Char data type is used to store any character.

❏ Example . char letterA ='A'

b. Reference Data Types:

Java Literals:

A literal is a source code representation of a fixed value. They are represented directly in the code without any computation.

Literals can be assigned to any primitive type variable. For example:

byte a = 68;

```
char a = 'A'
```