CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CE251: Java PROGRAMMING

July – November 2020

Chapter – 1

Java Introduction

DEPSTAR
Devang Patel Institute of Advance Technology and Research

Importance of Java Programming in IT Industries.

Why is Java important?

What is the difference between C, C++ & Java?

What are the class elements of Java.

How many Keywords of Java?

Classification of Java Keywords

1. Data Types
2. Flow Control Statement
3. Exception Handling
4. Class Level
5. Object Level
6. Method
7. Modifiers
8. Un Used

Classification of Java Keywords

1. Data Types - 8
2. Flow Control Statement -10
3. Exception Handling – 5 (try, catch, finally, throws, throw)
4. Class Level – 6 (class, extends, interface, implements, package, import)
5. Object Level – 4 (new, this, super, instanceof)
6. Method – 2 (void, return)
7. Modifiers – 11 (synchronized, strictfp, nativ, transient, volatile, abstract, final, static)
8. Un Used- 2 (goto, const)
9. Java8 – 2 (enum, assert)

Reserved Words in Java

boolean
byte
char
short
int
long
float
double
void

false
null
true

abstract
final
native
private
protected
public
static
synchronized
transient
volatile

break
case
catch
continue
default
do
else
finally
for
if
return
switch
throw
try
while

class
extends
implements
interface
throws

import
package

instanceof
new
super
this

byvalue
cast
const
future
generic
goto
inner
operator
outer
rest
var

reserved for future use.

How many types of Java Methods?

How many types of Java Classes?

Types of classes in java

1. Normal classes
2. Abstract classes
3. Final classes
4. Strictfp classes
5. Immutable classes
6. Mutable classes
7. Single ton classes

Steps to Design First Application in Java

1. Select the editor
2. Write the application
3. Save the application(.java)
4. Compiling application(javac file_name)
5. Execute the application(java file_name)

Difference between IDE & Editor?

Write the application

```
class Demo
{
    public static void main(String[] args)
    {
        System.out.println("Hello Java");
    }
}
```

What if in a source file multiple classes then how to save the application?

How to save this code?

```
class Demo
{
    public static void main(String[] args)
    {
        System.out.println(“Hello Java”);
    }
}
class A{
}
class B{
}
```

What if the source file contain **public** class?

How to save this code?

```
public class Demo
{
    public static void main(String[] args)
    {
        System.out.println(“Hello Java”);
    }
}
class A{
}
class B{
}
```

What about this source file?

```
class Demo
{
    public static void main(String[] args)
    {
        System.out.println("Hello Java");
    }
}
public class A{
}
class B{
}
```

Can we save the application by any name?

Rule

1. Public class-------→ save source by same name
2. No public class---→ save by any name

Compilation (javac)

What are the responsibility of the javac compiler?

Two action

1) Check the syntax error

2) .java ----? .class

How many **.class** will be generated if you save application as **Sample.java**?

```
class Demo
{
    public static void main(String[] args)
    {
        System.out.println("Hello Java");
    }
}
class A{
}
class B{
}
```

Execution (java)

What are the responsibility of the execution process?

Two action

1) Load .class file into the memory. If .class file is not found then raise an error

2) JVM call to main() method to start execution

Any Question??

Enhancement In First Application

1. How many packages in java?
2. Which one is optional package? & Why?

Basic Data Types in Java

What is the purpose of Data Type?

Three purpose

1. Data Type is representing type of variable
2. Data Type is representing/deciding memory size of variable
3. Data type decide range value of variable

Data Types in Java

Java has 8 fundamental data types.

The eight types are: byte, char, short, int, long, float, double, and boolean.

Fundamental Data Types

- All primitive types in Java have a defined size (in bits). This is needed for cross platform compatibility.
- Each type has a defined set of values and mathematical behaviour.
- Six of the types are numeric (byte, short, int, long, float, double) The *char* type holds characters
- The *boolean* type holds truth values

Integral Data Types

- 4 types based on integral values: byte, short, int, long
- All numeric types are signed. There are NO unsigned types in Java.
  - Integrals are stored as 2's compliment.

Type	Size	Range
byte	8 bits	-128 through +127
short	16 bits	-32768 through +32767
int	32 bits	-2147483648 through +2147483647
long	64 bits	-9223372036854775808 through +9223372036854775807

Floating point Data Types

- 2 types based on floating point values: float and
- double Storage conforms to IEEE 754 standard
- Floating point numbers are not accurate.They are an approximation
- floats store 7 significant digits. doubles store 15.

Type	Size	Range
float	32 bits	-3.4 * 10^{38} through +3.4 * $1^{3}0^{8}$
double	64 bits	-1.7 * 10^{308} through +1.7 * $1^{3}0^{08}$

Character data type

- The char type defines a single character
- In many other programming languages, character types are 8-bits (they store ASCII values). In Java, character types are 16-bits.
  Java characters store characters in *unicode* format.
- Unicode is an international character set which defines
- characters and symbols from several different world languages.
  - Unicode includes ASCII at its low range (0-255)
- Characters can be converted to integers to perform mathematical functions on them.

Boolean data type

- The boolean type defines a truth value: true or false.
- booleans are often used in control structures to represent a condition or state.
- Java characters store characters in *unicode* format.
- Unicode is an international character set which defines characters and symbols from several different world languages.
  - Unicode includes ASCII at its low range (0-255)
- booleans CANNOT be converted to an integer type.

Java Coding Convention

Classify into 3

1. Classes & Interface
2. Method & Variable
3. Packages & Constant

Difference b/w User define & Predefine

Java Identifier

Every user define name in java is called identifier

While declaring identifier having some rules

What are the rules?

Rule-1

The identifiers can contain

a. Lowercase a-z
b. Uppercase A-Z
c. Number 0-9
d. Underscore(_)
e. Dollar ($)

Rule-2

It should not starts with number

int abc123=10;

int _$ =20;

int 123abc= 30;

int abc.123 = 40;

Rule-3

- It is not possible to declare keywords as a identifier

Rule-4

- It is possible to use predefine class names & interface name as an identifier nut its not recommended

Rule-5

There is not length limit for identifiers but it is always to recommended to take less then 16 characters

Rule-6

- Java is case sensitive language so java identifiers is also case sensitive

Java Comments

What is the purpose of comments in programming languages?

3 Types of Comments in Java

1. Single Line
2. Multiline
3. Documentation

Java Variables

How many types of variables?

What is the purpose of variables?

3 types of variable in java

1. Local Variables
2. Instance Variables
3. Static Variables

Few points which we will discuss with respect to all types of variables

1. Where we declare
2. Scope of variable
3. Memory allocation
4. Memory destroyed
5. Initial values
6. How to access variable
7. Relation with object
8. Stored memory

Every variable must have some specific types

1. Primitive type
2. Class type
3. Array type

1. Local Variables

```
class Local
{
    public static void main(String[] args)
    {
        int a =10;
        int b = 20;
        System.out.println(a+b);
    }
}
```

So variables which declared inside the method is called local variable

1. Local Variables

What is the scope of variable?

Ans- inside the method only

1. Local Variables

When the memory will be allocated and destroyed?

```
void method()     // method start memory allocated
{
    int a =10;
    int b=20;
}                         // method end memory destroyed
```

1. Local Variables

Where it stored?

Ans- it stored in stack memory

1. Local Variable- Default value

```
class Local
{
    public static void main(String[] args)
    {   int a ;
        int b;
        ;;;;;;;;;;
        ;;;;;;;;;;
        a=100;
        b=200;
        S.O.P(a);
        S.O.P(a);

    }
    }
```

JVM will not assign default value, initialize before using

2. Instance Variable

Java contain two types of area generally

1. Instance area
2. Static area

```
void method()
{
    int a =10;
    int b=20;
}
```

```
static void method()
{
    int a =10;
    int b=20;
}
```

2. Instance Variable

```
class Instant
{
        int a =10;
        int b = 20;

    public static void main(String[] args)
    {
        System.out.println("hello I m in static area");
    }
    void m1()
    {   System.out.println("hello I m in instant area");
    }
}
```

Variables declared inside the class but outside of the method is called instance variable

2. Instance Variable

What is the scope of the variable?

Ans- Inside the class all the methods are able to access

2. Instance Variable

Memory allocation & destroyed

When object is created & destroyed

2. Instance Variable

Access permission of instance variable & method

Instance to instance ---? direct access

Instance to static ------? access throw object

2. Instance Variable- example

```
class Instant
{
        int a =100;
        int b = 200;

    public static void main(String[] args)
    {
        Instant  i = new Instant();
        S.O.P(i.a);              S.O.P(i.b);
    }
    void m1()
    {     S.O.P(a);              S.O.P(b); }
}
```

2. Instance Variable- Default value

```
class Instant
{
        int a;
        boolean b;

    public static void main(String[] args)
    {
        Instant  i = new Instant();
        S.O.P(i.a);              S.O.P(i.b);
    }
    void m1()
    {    S.O.P(a);              S.O.P(b); }
}
```

JVM will assign default value

2. Instance Variable

Where it stored?

Stored in Heap Memory

3. Static Variable

```
class StaticVariable
{
        static int a =10;
        static int b = 20;

    void m1()
    {   System.out.println("hello I m in instant area");
    }

    public static void main(String[] args)
    {
        System.out.println("hello I m in static area");
    }
    }
```

Variables declared inside the class but outside of the method with static modifier is called static variable

3. Static Variable

What is the scope of the variable?

Ans- Within the class all the methods, constructors & blocks are able to access

3. Static Variable

Memory allocation & destroyed

Static variable memory is allocated when .class file is loaded into the memory

3. Static Variable

Access permission of static variable & method

static to instance ---□ access by using class name

static to static ------□ access by using class name

3. Static Variable- example

```
class StaticVariable
{
        static int a =100;
        static int b = 200;

    void m1()
    {   S.O.P(StaticVariable.a);
        S.O.P(StaticVariable.b);

    }

    public static void main(String[] args)
    {
        S.O.P(StaticVariable.a);
        S.O.P(StaticVariable.b);
    }
    }
```

Will it be called to m1() method?

3. Static Variable- correct

```
class StaticVariable
{
        static int a =100;
        static int b = 200;

    void m1()
    {   S.O.P(StaticVariable.a);
        S.O.P(StaticVariable.b);

    }

    public static void main(String[] args)
    {
        S.O.P(StaticVariable.a);
        S.O.P(StaticVariable.b);
        StaticVariable s = new StaticVariable;
        s.m1();


    }
    }
```

3. Static Variable

Where it stored?

Stored in Non-Heap Memory

3. Static Variable- Default value

```
class StaticVariable
{
        static int a;
        static boolean b;

    public static void main(String[] args)
    {
        S.O.P(StaticVariable.a);
S.O.P(StaticVariable.b);
    }
    }
```

JVM will assign default value

Based on 3 types of variables complete the code

```
Class Test
{
//Create 2 instance variables
//Create 2 static method and print instance variable inside the method
// Call the above 2 static methods inside main method
}
```

2

```
Class Test
{
//Create 2 instance variables
//Create 2 static variables
//Create 1-static method, 1-instance method and print all 4 variable inside the method
// Call the above 2 static methods inside main method
}
```

3.

Class Student
{
//Create 3 instance variables
roll_no, marks, name

//Create 1 static variable
college_name and initalize it with "CHARUSAT"
}

Class StudentInfo
{
//Create object of Student class
// initialize all the values
// print all values for 2 different student
}

Difference b/w Instance & Static variable

Prepare the table

Any Question