OOP Labsheet-1

Prof.R.Gururaj

Genesis of Java:

C language- 1970 → 1989; C++ 1979 → 1983

Java : Sun Micro Systems (by James Gosling and four others)

Oak(1991-95) → Java (1996)

Java & Internet

Java buzzwords

- Simple
- Secure
- Portable
- Object-oriented
- Robust
- Multithreaded
- Interpreted
- Distributed

Java Keywords

abstract	boolean	break	byte	case
catch	char	class	const	continue
default	do	double	else	extends
final	finally	float	for	goto
if	implements	import	instaceOf	int
interface	long	native	new	
package	private	protected	public	
return	short	static	super	switch

synchronized	this	throw	throws	transient
try	void	volatile	while	
true	false	null		

In the latest version of JDK we have more than 50 keywords

Three important Principles of Java

Encapsulation- is the mechanism that binds together code and the data it uses, and keeps both protected from outside interference and misuse.

I***nheritance***- Is the process by which one object acquires the properties of another.

Polymorphism-One interface for a general class of actions.

Simple Java program and its working

```
class FirstJavaProgram
{
    public static void main(String args[])
    { System.out.println("This is my First Java Program");}
}
```

> javac FirstJavaProgram.java

> java FirstJavaProgram

Output: This is my First Java Program

Class: The class forms the basis for object oriented programming in Java. Class is a template for an object, and object is an instance of a class.

General form of a class:

class *classname*

{

type instance-variable1;

type instance-variable2;

….

type instance-variableN;

type methodname1(parameter-list)

{ body }

……

type methodnameN(parameter-list)

{ body }

}

Members of a class

instance variables

methods

A simple class:

class *Box*

{

int width;

int length;

int depth;

}

To instantiate an object of Box

Box myBox=new Box();

To access variables of an object

myBox.length=12;

Java Identifiers- can contain all characters, numeric chars, in any case, '$',' _'

but can't start with a numeric character, '-'(hyphen), '/' not allowed

Valid: length, length2, box_Length, box$length etc.

Invalid: 2length, my-box, box/length etc.

Java Control statements

'if' conditional statements

1) **if statement**

if (condition) statement;

2) **if - else statement**

if (condition) statement;

else statement;

if - else if - else

if (*condition*) *statement;*

else if*(condition) statements;*

else if*(condition) statement;*

…….

else *statement;*

for loop

for (*initialization; condition; iteration*)

{ body

}

switch - case

int month;

. . .

switch (month) {

case 1: System.out.println("January"); break;

case 2: System.out.println("February"); break;

case 3: System.out.println("March"); break;

default: System.out.println(“April and beyond");

}

while loop

while(expression){

statements

}

do – while loop

do {

statements

} *while* (booleanExpression);

Java Data Types

Type	*Size/Format*	*Description*
		(integers)
byte	8-bit	Byte-length integer
short	16-bit	Short integer
int	32-bit	Integer
long	64-bit	Long integer
		(real numbers)
float	32-bit	Single-precision floating point
double	64-bit	Double-precision floating point
		(other types)
char	16-bit Unicode character	A single character
boolean	true or false	A boolean value (true or false)

Java Literals

- ❑ Integer literals
- ❑ Floating-point literals
- ❑ Boolean literals
- ❑ Character literals
- ❑ String literals

Java Type conversion

- ❑ If two types are compatible, then Java will perform the conversion automatically.

- Ex: assign an *int* value to *long* *//widening*
- However not all types are compatible.
- Ex: Conversion from double to byte.
- We need to do casting for this conversion of incompatible types.

Java automatic Type conversion

It is done when following two conditions are met.

1. Two types are compatible
2. Destination type is larger than the source

Ex: Byte to int.

There is no auto conversion numeric types to char or Boolean.

Java automatic Type Promotion

byte a=40;

byte b=50;

byte c=100;

int d= a*b/c;

(byte/short/char)→ int → long → float →double

Single-dimensional Arrays:

```
int[] arrayOfInts;
int[] arrayOfInts = new int[10];

elementType[] arrayName = new elementType[arraySize];
String[] arrayOfStrings = new String[10];
```

```
int intarray[]={2,5,6};
```

Two dimensional

```
int marks[] [] = new int[3][4];
```

```
//some Examples on arrays//
```

Taking input from the user:

Exercise Problems:

Exercise 1*:* Write a Java program to convert temperature from Fahrenheit to Celsius degree.

Test Data
Input Fahrenheit value 212
Expected Output
Equivalent Celsius value 100

Exercise 2: Write a Java program to display ASCII value of a character.

Test Data

Input character: b

Expected Output

ASCII value: 98

Exercise 3: Write a Java program to display the Diameter, Circumference and Area of a circle.

Test Data

Radius value: 6

Expected Output

Diameter of a circle : 12.0

Circumference of a circle: 37.68

Area of a Circle: 113.04