

Department of Computer Science

Gujarat University



Certificate

Roll No: 30

Seat No: _____

This is to certify that Mr./Ms. Rathod Ajinkya Sreekant
student of MCA Semester – III has duly completed his/her term work for
the semester ending in December 2020, in the subject of _____
Java Programming towards partial fulfillment of his/her Degree of
Masters in Computer Applications.

Date of Submission
12 - December - 2020

Internal Faculty

Head of Department

Department Of Computer Science Rollwala Computer Centre
Gujarat University

MCA - III

Subject: - Java Programming

Name: - Ajinkya Rathod

Roll No.: - 30

Exam Seat No.: - _____

Name: Ajinkya Rathod

Semester: III

Subject: Java

Q1.

The Bytecode:

- Bytecode is highly optimized set of instructions designed by the Java runtime system which is called JVM.

JVM is called Java Virtual Machine

Translating a Java program into bytecode makes it much easier to run a program in wide variety of environments because only JVM needs to be implemented for each platform.

At Compile Time

Source code → Compiler → Bytecode

Native
Machine
Code.

← JIT
Compiler

Q2. Features of Java

1. Simple

Java is easy to learn and its syntax is simple, clean and easy to understand.

2. Object - Oriented

In java, everything is an object which has some data and behaviour.

Java can be easily extended as it is based on Object Model.

Concepts of OOP:-

→ Object

→ Class

→ Inheritance

→ Polymorphism

→ Abstraction

→ Encapsulation.

3) Robust

Java makes an effort to eliminate error-prone codes by emphasizing mainly on compile-time error checking and runtime checking.

But the main areas where Java improved Memory Mgt. and mishandled Exceptions by introducing automatic Garbage Collection and exception handling.

4) Platform Independent

Unlike other languages, C++ which are compilable into platform specific machine, Java is guaranteed to be write-once run anywhere language.

5) Secure

Java secure features enable us to develop virus free, temper free system.

6.

Multi-Threading

Multi Threading enables us to do many of tasks simultaneously. Its advantage is that it utilises same memory and other resources to execute multiple threads at once.

7.

Portable

Java's Byte Code can be carried to any platform. No implementation dependent features.

8.

High Performance

Java is between compiler and interpreter. Java's high performance is cause of JIT Compiler.

Q3.

How applets changed the internet.

Ans

Security

Every time you download a "risky" code you are taking risk because you code that downloaded might contain a virus, Trojan horse or anything.

Java achieved this protection by confining an applet to Java execution environment and not all parts to be downloaded.

Portability

It is a major aspect of Internet because there are many OS connected to it.

If a Java code were to run on virtually any computer, there needed to be some way to enable that program to execute on various

systems.

It is not practical to have all versions of applet for all computers. The same code must work on all computers.

Therefore, some means of generating portable executable code was needed.

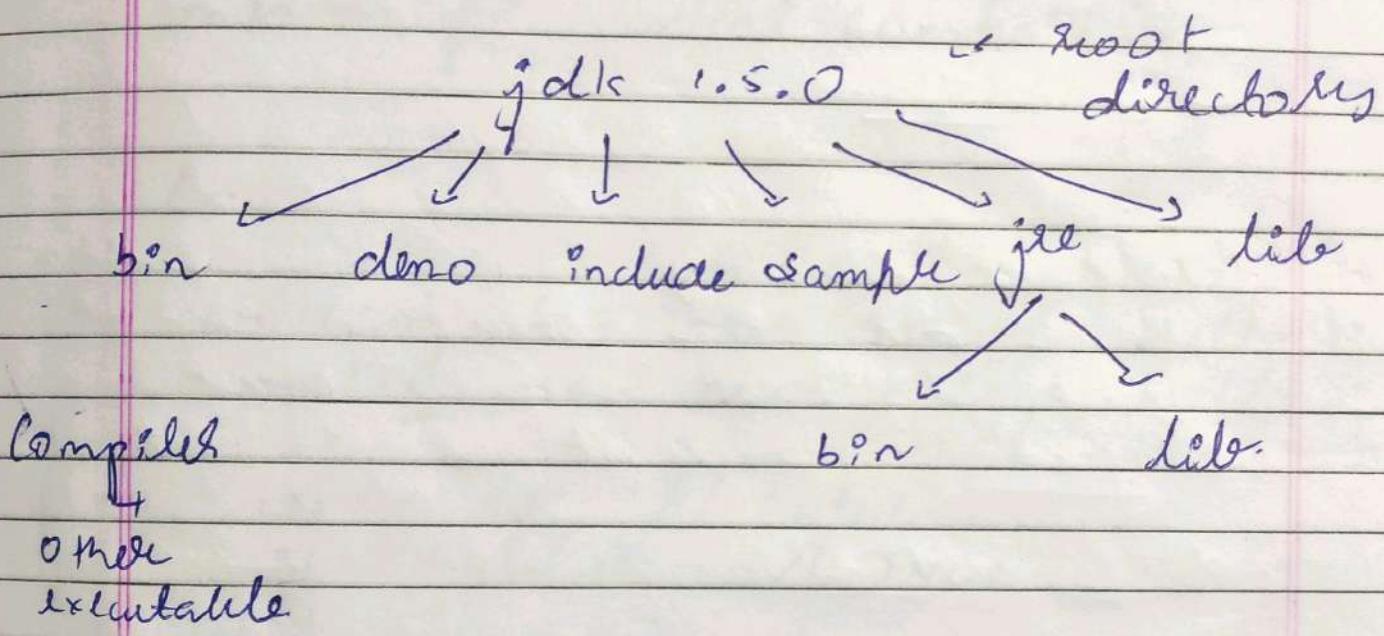
Jinky
MCA - III

Rathod.

Java

30

Q4. Explain JDK root directory with diagram.



→ jdk 1.5.0 is referred to as the root directory for Java.

→ The sample directory contains the sample application that uses SNLP, which is Java networking lang protocol.

ALL INFORMATION IS FOR INTERNAL USE ONLY
STORING OR REPRODUCING IN WHOLE OR IN PART IS PROHIBITED

NEW INDIA
PAGE NO.: DATE

- JRE directly contains Java runtime facilities that are used when you execute a Java program

Data-type, variable constants and loops

NEW INDIA
PAGE NO.: DATE

Q:

Difference between '==' and equals
in java.

- .equals() is a method and "==" is a operator.
- we can use "==" operator for reference comparison and .equals() method for content comparison.

Ex:-

```
String s1 = new String("Hello");
String s2 = new String("Hello");
```

```
s1 == s2 // False
s1.equals(s2) // True
```

Q. Discuss & types of byte according with example & state difference between two.

→ The two types are:-

- Big Endian
- Little Endian.

Big Endian

Constant denoting big endian byte order. In this order, multi-byte values are read from most significant to least significant.

Little Endian

Multi-byte values are ordered from least significant to most significant.

Eg.

In 32 bit, value of 45678912

Address	00	01	02	03
Little Endian	12	98	67	45
Big Endian	45	67	98	12

Q. Why do things are immovable

→ These objects are particularly useful in concurrent applications since they cannot change state, they cannot be corrupted by thread interference or destroyed from an inconsistent state.

→ The objects they refer to their address is never changed. For instance they point to some immutable.

Q: Final, Finally, Finalize

(i) Final

- > It is a keyword
- > It is used to apply restriction on class, method and variable. Final class can't be inherited.
- > It cannot be overridden and final variable value can't be changed

(ii) Finally

- > It is a block.
- > It is used to hold important code whether exception is handled or not.

(iii) Finalize

It is a method.

→ It is used to perform clean up processing just before garbage collected.

Q Static Method, variables and classes.

Q Static variable

A single copy of variable is created and shared by among all objects of that class.

Static Method

They do not use any instance.

They take all data from parameters and complete something from those parameters with no reference to variable.

Static Class

A class can be made static only if it is static class. Normal class don't need a reference of class.

Generics

- Q. Discuss benefit of generic over non-generic types.
- Ans. It enables types to be parameters when defining classes, interfaces and methods.
- The difference is that the inputs to formal parameters are type. Input to type parameters are type.
- Code that uses generic has many benefits over non-generic code.
 - Stronger type checks at compile time.
 - Elimination of casts.

The following are skipped as part of generic requires casting.

```
List lst = new ArrayList();
```

```
lst.add("Hello");
```

```
String s = (String) lst.get(0);
```

return the object so we have to convert
it to string type.

When re-written to use generics, no
codes does not require casting

```
List<String> lst = new ArrayList<String>();
```

```
lst.add("Hello");
```

```
s = lst.get(0);
```

3) Enabling programmers to implement generic
algorithms.

Q: What is issue with sequence w/
generics.

A: Java compiler does not actually
check different version of generic or
any other generic does.

short note on

1. Generic Constructors

A constructor can be generic even if its class is not.

Ex.: In following program the class sum() is not generic, but its constructor is.

class summation {

 private int sum;

 CT extends Number summation

 (CT args)

}

sum=0;

for (int i=0; i<args.length;

 sum+=i;

; i++)

}

int getSum() {

 return sum;

}

}

class avg(Ans Docx)

public static void main(String[] args)

Sunario ob = new Sunario(5.0)

System.out.println(ob.getScore());

}

3

- Recall that summation() is son of all the whole ~~to~~ numbers like 0 and N.
- Because summation() specifies a type parameter to be stored by name, a sum object can be created using any numeric type.
- Therefore, it is not necessary for a class summation to be generic, only a generic constructor is needed.

2) Generic Interface

- An interface can be generic
compulsory CTs were used to
ensure that two arrays could be
compared.
- They all specified just like generic
classes

Syntax

interface InterfaceName<type - polar - list>

Example

Interface contains net CTs &
localon contains (T0);

)

class myclass {
 T[] arrayRef;

myclass(T[] o) {
 arrayRef = o;

public void contains(T o) {

for (T x : arrayRef) {
 if (x.equals(o)) return true;
}

return false;

}

Q3

Generic Functional Interface

A lambda expression can't specify type parameters, so it's not generic.

However generic interface combined with lambda expression is generic.

In this case, the exact input of lambda expression has determined by type of delegate split for whom function implementation is divided.

Syntax

Interface somefunc λ
 $T \rightarrow \text{func}(T, t)$

}

interface myfunc $\langle T \rangle \lambda$
 $T \text{ comake}(i, t);$

}

public class generic Enterprise 2

public static void main(String[] args){

 my generic string > copy(size) ->

 String result = "";
 for (int i = size - 1; i >= 0; i--) {
 result += str.charAt(i);
 }
 return result;
 }

}

}

}

Q Generic Method

At all methods that include
this own type parameter. This is
similar to defining a generic type
but the type parameter's scope is
limited.

This syntax for generic method includes

a list of type parameters inside
angular brackets, which appear
where we methods return type.
'Optional section' must appear before
methods return type.

The actual class includes a generic
method compare which compares two
parallel algorithms.

public class List2

public static <T> boolean
compare(Pair<T, T> p1,
 T, T> p2)

return p1.getLValue().equals(
 p2.getLValue())
&& p1.getRValue().equals(
 p2.getRValue());

} 3

public class pair<K, V> {

private K key;
private V value;

public pair(K key, V value) {
this.key = key;
this.value = value;}

}

public void setKey(K key) { this.key = key; }

public void setValue(V value)
{ this.value = value; }

}

public K getKey() {
return key; }

public V getValue() {
return value; }

)

Pair (int, str)

p1 = new pair (2, "Apple");

Pair (int, str)

p2 = new pair (3, "Peaches");



Comparators
Eg: Lambda
Eg: Functions

Q: What are comparators and comparable.

→ Comparable

It is an interface which defines a way to compare an object that have self dependency to sort themselves.

→ The object must known how to order themselves

Eg: Roll no
Age
Salary, etc..



PAGE NO.

DATE:

* Comparable

A comparable interface is used to order the objects of specified class.

This is found in `java.util package`.

- > It contains two method
 - `Compare(Object obj1, Object obj2)`
 - `equal(Object obj)`

- > The first method compare object `obj1`, `Object obj2`.

Compare is two input objects & show come the output.

It takes a hexaidecimal integer 2013
as a positive integer to show
whether the first object is less than
that is greater than the second



-> The second method ~~method~~ ^{returns} (Object client) requires an object as a parameter (shows ? + V input after is each the parameter).

This will return the, only 1 action object is also complete.

- 1. List client & provided function
- 2. Interfaces
- Functional Interface

-> In java, an interface contains only one single abstract method.

-> A functional interface can contain default & static methods which do have an implementation.

Pre-defined Functional Interface

In many cases, however you will need to define your own functional interface because ODK8 and a new package called `java.util.function`



that provides several predefined
ops.

Interfaces that are predefined

1. Unary Operator $\leftarrow T \rightarrow$

Apply a unary operation to an object
of type T & return one result
which is also of type T , Its method
is called 'apply'.

2. Binary operator.

3. Comma

4. Suffix

5. Prefix

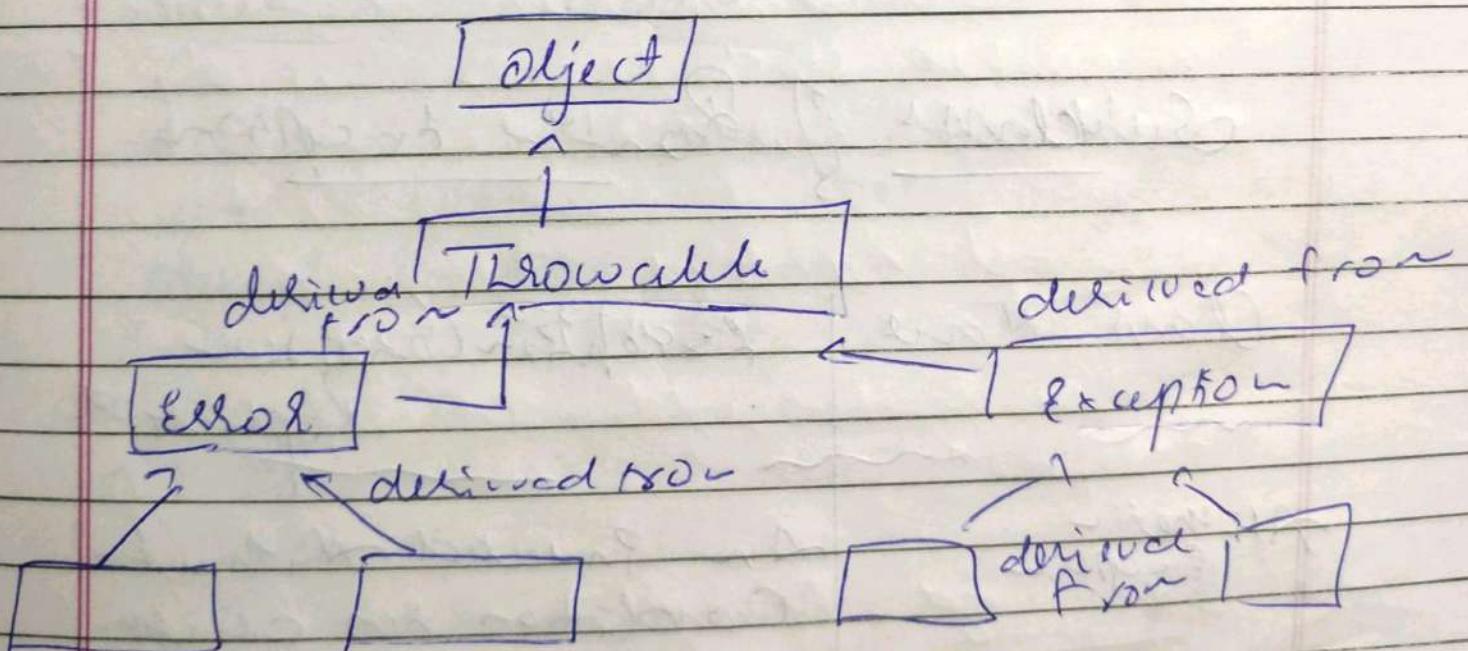
Unit - 7

Exception

O1 Discuss Exception & types of exception in Java.

A1 Two direct subclasses of class Throwable are class Error & class Exception - Only all standard exception.

Misery is which these classes belong.



Exception you
should not catch



Runtime Exception

- They are treated differently because of serious error in your code.
- Quite a lot of subclass of Runtime exception are used to signify problem in various packages in Java class library.

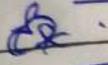
Subclasses of Runtime exception

Class Name	Exception Condition Reprinted
ArithmeticException	An invalid arithmetic condition has occurred such as zero division etc.

An invalid arithmetic condition has occurred such as zero division etc.



Index Out of Bound Exception



To use an index that is outside the bounds of object it is applied to. This may be an array, as the object or vector object

Negative Array Size Exception

To define an array with a Negative dimension

Null Pointer exception

An object variable contains null is used when it should refer to an object for proper operation

All arrays exception

To store an array that is permitted for array + \oplus

Q2

Explain Commonly used methods defined by Throwables

→ All exceptions are instances of Throwables so all exception contains the methods defined by Throwables.

→ Commonly used method defined by Throwables

→ Throwables.fillInStacktrace()

Returns a Throwable object that contains a complete stack trace. This object can be thrown.

→ String getLocalizedMessage()

Returns a localized description of exception

→ String getStackTrace()

This returns an array of strings describing the current exception. This will typically be fully qualified.



name of exception class.

4) void print Stack Trace()

This will output the message & the stack trace of standard error input stream which is seen in case of console program.

5) void print Stack Trace (ostream&)

This is same as previous method except that can specify the output stream as an argument earlier method for an exception object is available to

6. void print Stack Trace (System.out);

7. void print Stack Trace

Sends the stack trace to specific stream

Unit - 10

Java - I/O - Package

① Byte Stream

Byte Streams provide a convenient means for reading input & output of bytes.

1. They are used for reading & writing binary data.
2. All byte streams are descendants from Input stream & Output stream.
3. E.g. Images, sounds etc..

② Character Stream

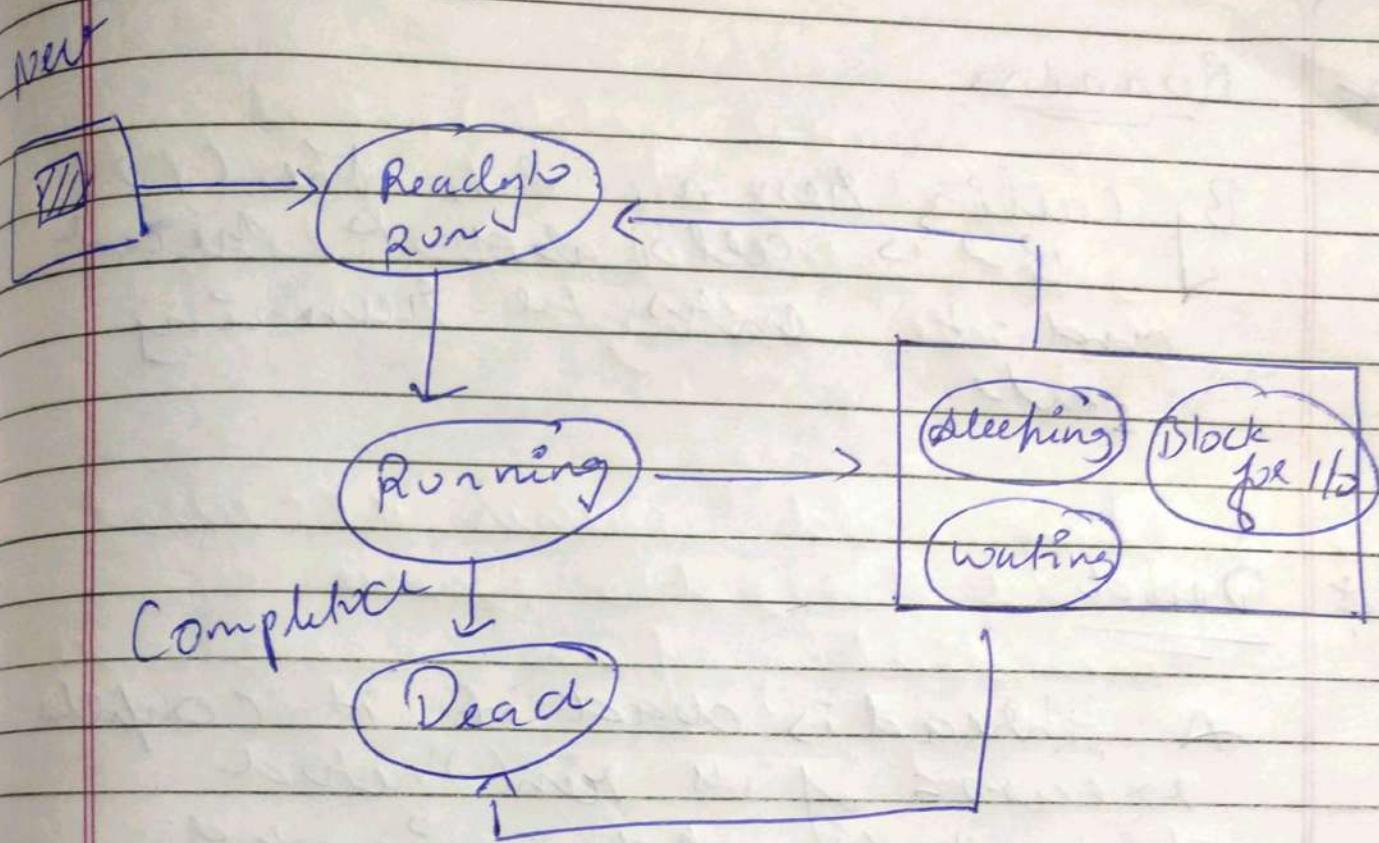
They are designed for handling the input and output of characters.

Thread

Q Explain Thread life cycle with Diagram

- A thread passes through several states throughout its lifetime & after completing its task it is dead
- Different states of thread are as follows

- 1) New
- a) Ready to Run
- 3) Running
- 4) Block
- * 5) Dead



New

At the time of thread creation, it is new state. By calling `start()` method. The thread will start its execution.

Ready to Run

After `start()` method, the thread is ready to run state. It means that thread is ready for CPU allocation.

* Running

By calling `ren() method`, the CPU
↓
is allocated to thread
and it starts the running
state.

* Dead

A thread is dead if it completes
execution of its `run()` method
or its `stop()` method is invoked.

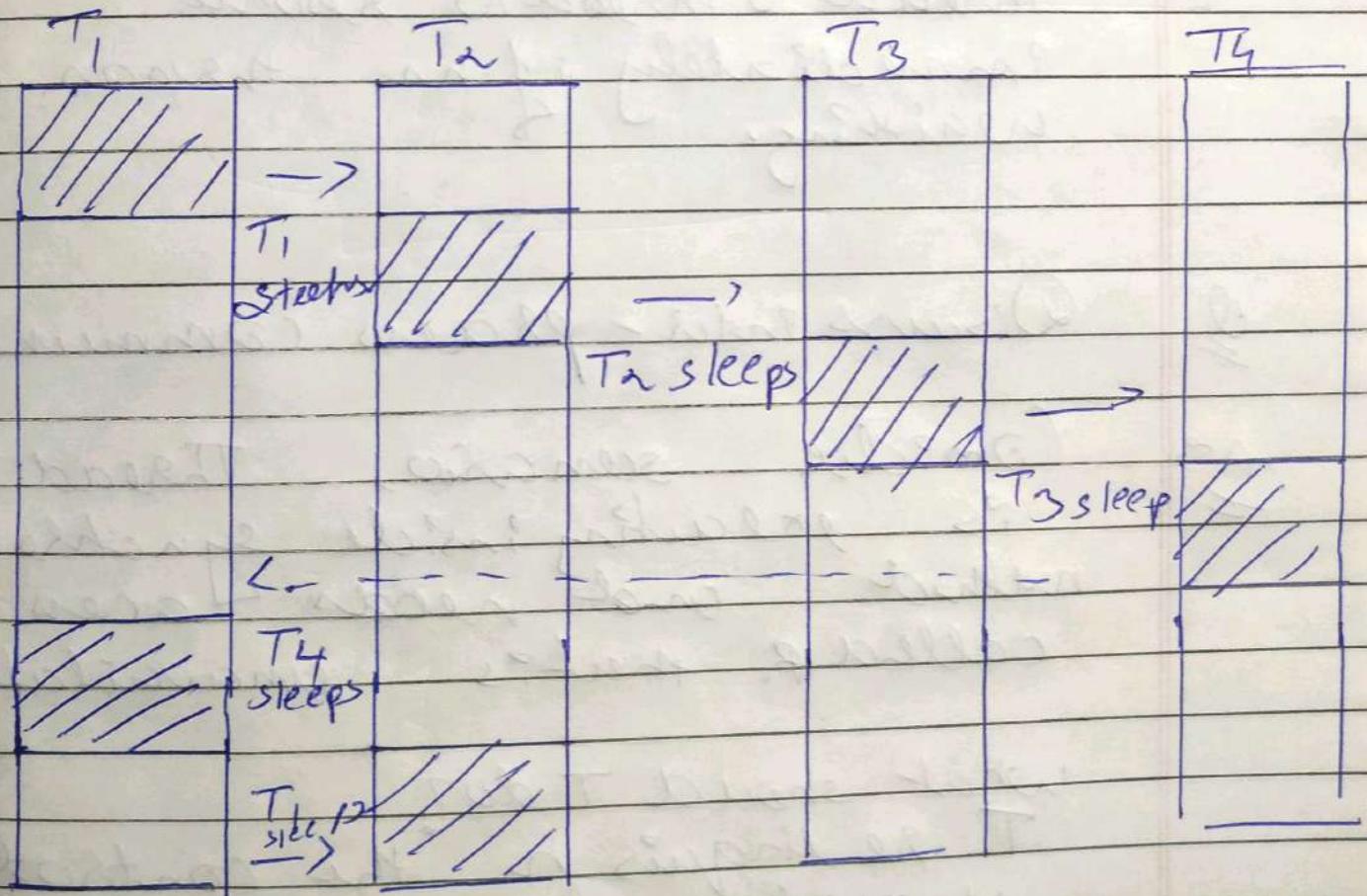


PAGE NO.

DATE:

Discuss Thread Scheduling

The scheduling of thread depends on your operating system some extent, but each thread will certainly get a chance to execute with other threads all active



PAGE NO. _____
DATE: _____

Their another `yield()` method, defined in `Thread class`, that gives other thread to have look, if they are waiting.

- Calling `yield` causes the current thread's resource immediately if no threads are waiting.

Q. Discuss inter-process communication.

Ans
Consider scenario, Thread T is executing inside synchronized method and needs access to called R matrix unavailable.

What should T do?

T relinquishes the control of object, allowing another thread to run. When R becomes available, T can be notified and resume execution.



Some form of interthread
(Communication) in which
one thread can notify
another is blocked and can
be notified that it can resume
execution.

- > The `wait()`, `notify()` and `notifyAll()` methods are part of all objects because they are implemented by Object class
- > When thread is temporarily blocked from running, it calls `wait()`.
- > A call to `notify()` resumes thread with all highest priority, `notifyAll()` gives access to Object.

**DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. - III**

ROLLNO : 30

NAME : Ajinkya Rathod

S U B J E C T : Java Programming

Hands on: Classwork - 1

```
/*
=====
=====
*
* Roll No: 30
*
* File: 01-Hello.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
=====
===== */

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

```
/*
=====
=====
Output:
```

Hello World

```
=====
===== */
/*
=====
=====
*
* Roll No: 30
*
* File: 02-CLA.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
=====
===== */

===== */
```

```

class CLA{
    public static void main(String[] args){

        // Prints length of total arguments
        int argsCount = args.length;
        System.out.println("Total arguments: " + argsCount);

        // Iterates over every argument
        for(int i = 0; i < argsCount; i++)
            System.out.println("Argument no " + i + " is: " + args[i]);

    }
}

```

/*

=====
=====

Output:

java CLA 10 20 30 40 50

Total arguments: 1

Argument no 0 is: 10

=====
=====

===== */

/*
=====

=====
=====

*

* Roll No: 30

*

* File: 03_04-Arithmetic.java

* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)

*

*

=====
=====

===== */

class Arithmetic{

public static void main(String[] args){

System.out.println("N1: 25, N2: 10\n");

```
int a = 25;
int b = 10;

/* Arithmetic Operations on Integer */
System.out.println("Arithmetic Operations on Int");

// Addition
int add = a + b;
System.out.println("-Addition: " + add);

// Subtraction
int sub = a - b;
System.out.println("-Subtraction: " + sub);

// Multiplication
int mul = a * b;
System.out.println("-Multiplication: " + mul);

// Division
int div = a / b;
System.out.println("-Division: " + div);

// Modulo Division
int mod = a % b;
System.out.println("-Modulo: " + mod);

// Unary Minus
int minus = -a;
System.out.println("-Unary Minus of N1: " + minus);

/* Arithmetic Operations on Float */
System.out.println("\nArithmetic Operations on Float");

float x = 25f;
float y = 10f;

// Addition
float addF = x + y;
System.out.println("-Addition: " + addF);

// Subtraction
float subF = x - y;
System.out.println("-Subtraction: " + subF);

// Multiplication
float mulF = x * y;
```

```

System.out.println("-Multiplication: " + mulF);

// Division
float divF = x / y;
System.out.println("-Division: " + divF);

// Modulo Division
float modF = x % y;
System.out.println("-Modulo: " + modF);

// Unary Minus
float minusF = -a;
System.out.println("-Unary Minus of N1: " + minusF);
}

}

/*
=====
=====
Output:

```

N1: 25, N2: 10

Arithmetic Operations on Int
 -Addition: 35
 -Subtraction: 15
 -Multiplication: 250
 -Division: 2
 -Modulo: 5
 -Unary Minus of N1: -25

Arithmetic Operations on Float
 -Addition: 35.0
 -Subtraction: 15.0
 -Multiplication: 250.0
 -Division: 2.5
 -Modulo: 5.0
 -Unary Minus of N1: -25.0

```

=====
===== */
=====
===== *
* Roll No: 30

```

```

/*
* File:    05-Galloon-to-Litres.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
=====
===== */

class GalloonToLitres{
    public static void main(String[] args){

        // Value of Galloon
        double galloon = 85.12;

        // Converting Galloon to Litres
        double litres = galloon * 3.78541;

        // Printing Solution
        System.out.print(galloon + " Galloon/s = ");

        // Setting Precision to 2 digits after decimal
        System.out.println(String.format(" %.2f", litres));
    }
}

```

```

/*
=====
=====
Output:

```

85.12 Galloon/s = 322.21

```

=====
===== */
=====
=====

* Roll No: 30
*
* File:    06-if-else.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
```

```

/*
=====
===== */
class IfElse{
    public static void main(String[] args){

        int number = 0;

        // If number is "0"
        if(number == 0){
            System.out.println(number + " is neither even nor odd");
        }

        // If number is Even
        else if (number % 2 == 0){
            System.out.println(number + " is Even");
        }

        // If number is Odd
        else{
            System.out.println(number + " is Odd");
        }

    }
}

```

```

/*
=====
=====
Output:

```

0 is neither even nor odd

```

=====
===== */
=====
=====
*
* Roll No: 30
*
* File: 07-Display-Sum-and-Max.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
```

```
*=====
=====
class Array {
    public static void main(String[] args) {
        double[] myList = {1.9, 2.9, 3.4, 3.5};

        // Printing All Elements
        for(int i = 0; i < myList.length; i++){
            System.out.println(myList[i] + " ");
        }

        // Adding All Elements
        double total = 0;
        for(int i = 0; i < myList.length; i++){
            total += myList[i];
        }
        System.out.println("Total: " + total);

        // Fetching Max
        double max = myList[0];
        for(int i = 1; i < myList.length; i++){
            if (myList[i] > max){
                max = myList[i];
            }
        }
        System.out.println("Max: " + max);

    }
}
```

```
/*
=====
=====
Output:
```

```
1.9
2.9
3.4
3.5
Total: 11.7
Max: 3.5
```

```
===== */ /*
=====
=====
*
* Roll No: 30
*
* File: 08-CharArithDemo.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
=====
*/
```

```
class CharArithDemo{
    public static void main(String args[]){
        // Assiging Character using Char
        char ch = 'A';
        System.out.println(++ch);

        // Assiging Character using Int
        ch = 65;
        System.out.println(++ch);
    }
}
```

```
/*
=====
=====
Output:
```

B
B

```
=====
=====
*/ /*
=====
=====
*
* Roll No: 30
*
* File: 09-sound.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
```

```

/*
=====
===== */
class Sound {
    public static void main(String args[]) {
        double dist;
        dist = 7.2 * 1100;

        System.out.println("The lightning is " + dist + " feet away.");
    }
}

```

```

/*
=====
=====
Output:

```

The lightning is 7920.0 feet away.

```

=====
===== */
=====
=====
*
* Roll No: 30
*
* File: 10-ScopeDemo.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
===== */
===== */

```

```

class ScopeDemo {
    public static void main(String args[]) {
        int x = 10;

        if(x == 10){
            int y = 20;

            System.out.println("x and y: " + x + "" + y);
            x = y* 2;
    }
}

```

```

        System.out.println("x is " + x);
    }
}

/*
=====
=====
Output:
x and y: 1020
x is 40

=====
*/ /*
=====
=====
*
* Roll No: 30
*
* File: 11-Boolean-LogicalOpDemo.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*/
===== */

class LogicalOpTable{
    public static void main(String args[]){
        boolean a,b;
        int p,q;

        a = true; b = true;

        // Setting a and b as numeric
        if(a) p = 1; else p = 0;
        if(b) q = 1; else q = 0;

        // System.out.println("P\tQ\tAnd\tOR\tXor\tNot\n");
        System.out.println("P\tQ\n");

        // -----
        System.out.println(p + "\t" + q + "\t");
        System.out.println((p&q) + "\t" + (p|q) + "\t");
        System.out.print((p^q) + "\t");
    }
}

```

```
if(!a){ System.out.println("1" + "\t");}
else{ System.out.println("0" + "\t"); }

System.out.println("");
// ----

a = true; b = false;
// Setting a and b as numeric
if(a) {p = 1;} else {p = 0; }
if(b) {q = 1;} else {q = 0; }

System.out.println(p + "\t" + q + "\t");
System.out.println((p&q) + "\t" + (q|q) + "\t");
System.out.print((p^q) + "\t");
if(!a){ System.out.println("1" + "\t");}
else{ System.out.println("0" + "\t"); }
System.out.println("");

// ----

a = false; b = true;
// Setting a and b as numeric
if(a) {p = 1;} else {p = 0; }
if(b) {q = 1;} else {q = 0; }

System.out.println(p + "\t" + q + "\t");
System.out.println((p&q) + "\t" + (q|q) + "\t");
System.out.print((p^q) + "\t");
if(!a){ System.out.println("1" + "\t");}
else{ System.out.println("0" + "\t"); }
System.out.println("");

// ----

a = false; b = false;
// Setting a and b as numeric
if(a) {p = 1;} else {p = 0; }
if(b) {q = 1;} else {q = 0; }

System.out.println(p + "\t" + q + "\t");
System.out.println((p&q) + "\t" + (q|q) + "\t");
System.out.print((p^q) + "\t");
if(!a){ System.out.println("1" + "\t");}
else{ System.out.println("0" + "\t"); }
```

```
    }
}

/*
=====
=====
Output:

P    Q

1    1
1    1
0    0

1    0
0    0
1    0

0    1
0    1
1    1

0    0
0    0
0    1

=====
===== */
===== */

*
* Roll No: 30
*
* File:    12-Table.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
===== */
===== */

class Table {
    public static void main(String[] args) {

        if(args.length > 1 || args.length < 1){
            System.out.println("Enter only 1 argument");
        }
    }
}
```

```

        System.exit(1);
    }

    int number = Integer.parseInt(args[0]);
    for(int i = 1; i <= 10; i++)
        System.out.println(number + " * " + i + " = " + (number * i));

}
}

```

```

/*
=====
=====
Output:

```

```

10 * 1 = 10
10 * 2 = 20
10 * 3 = 30
10 * 4 = 40
10 * 5 = 50
10 * 6 = 60
10 * 7 = 70
10 * 8 = 80
10 * 9 = 90
10 * 10 = 100

```

```

=====
=====
 */ /*
=====
=====
*
* Roll No: 30
*
* File: 14_17-BitDemo-and-Shift-Operators.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
=====
*/

```

```

class JavaOperators_Bit{
    public static void main(String[] args){
        // Bitwise Operators

```

```
int a = 65; // 100 000 1
int b = 12; // 000 110 0

System.out.println(a & b); //000 000 0
System.out.println(a | b); //100 110 1
System.out.println(a ^ b); //100 110 1
System.out.println(~a); //111 001 0

System.out.println();

// Bit Shift
int c = -10;
var d = -10L;
System.out.println(c << 2);
System.out.println(c >> 2);
System.out.println(c >>> 2); // Unsigned

System.out.println();
System.out.println(d << 2);
System.out.println(d >> 2);
System.out.println(d >>> 2); // Unsigned Right Shift
// There is no unsigned Left Shift
}

/*
=====
=====
Output:

0
77
77
-66

-40
-3
1073741821

-40
-3
4611686018427387901

=====
```

```
=====
 * / /*
=====
=====
 *
 * Roll No: 30
 *
 * File:    15-MathClass.java
 * Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
 *
 *
=====
 ===== */
import java.lang.Math;

class MathClass
{
    public static void main(String[] args)
    {
        int a = -10;
        float b = -20.20f;

        // Use of .abs() method to get the absoluteValue
        int AbsA = Math.abs(a);
        float AbsB = Math.abs(b);

        System.out.println("Absolute value of int: " + AbsA);
        System.out.println("Absolute value of float: " + AbsB);
        System.out.println("");

        double x = Math.PI;
        System.out.println("Math.PI: " + Math.PI);
    }
}

/*
=====
=====
Output:

Absolute value of int: 10
Absolute value of float: 20.2

Math.PI: 3.141592653589793
```

```
=====
===== */ /*
=====
=====
=====
*
* Roll No: 30
*
* File:    16-ArithmeticCharCode.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
===== */
===== */

class CharCodeCalcs {
    public static void main(String[] args){
        char letter1 = 'A';
        char letter2 = (char)(letter1+1);
        char letter3 = letter2;

        System.out.println("Here's a sequence of letters: "+ letter1 +
letter2 +
                (++letter3));

        System.out.println("Here are the decimal codes for the letters:\n"+
letter1 + ": " + (int)letter1 +
" " + letter2 + ": " + (int)letter2 +
" " + letter3 + ": " + (int)letter3);
    }
}

/*
=====
=====
=====
Output:

Here's a sequence of letters: ABC
Here are the decimal codes for the letters:
A: 65 B: 66 C: 67
=====
===== */
/*
```

```
=====
=====
/*
 * Roll No: 30
 *
 * File: 18-Binary-with-Indicators.java
 * Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
 *
 */
===== */
===== */

import static java.lang.Integer.toBinaryString;

class BinaryWithIndicators{
    public static void main(String[] args){
        int indicators = 0xFF07;
        int selectBit3 = 0x4;

        System.out.println("Indicators: " + toBinaryString(indicators));
        System.out.println("Indicators: " + toBinaryString(selectBit3));

        indicators &= selectBit3;
        System.out.println("indicators: " + toBinaryString(indicators));
        System.out.println("selectBit3: " + toBinaryString(selectBit3));

        indicators &= selectBit3;
        System.out.println("indicators: " + toBinaryString(indicators));
        System.out.println("selectBit3: " + toBinaryString(selectBit3));

        indicators = 0xFF09;
        System.out.println("indicators: " + toBinaryString(indicators));
        System.out.println("selectBit3: " + toBinaryString(selectBit3));

        indicators |= selectBit3;
        System.out.println("indicators | selectBit3: " +
toBinaryString(indicators));

        indicators &= ~selectBit3;
        System.out.println("Switched Off");
        System.out.println(toBinaryString(indicators));
    }
}
```

```

/*
=====
=====
Output:

Indicators: 111111100000111
Indicators: 100
indicators: 100
selectBit3: 100
indicators: 100
selectBit3: 100
indicators: 111111100001001
selectBit3: 100
indicators | selectBit3: 111111100001101
Switched Off
111111100001001

=====
*/ /*
=====
=====
*
* Roll No: 30
*
* File: 19-Methods-on-Operations-on-Bits.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
===== */
import static java.lang.Long.*;

class TryBitMethods {
    public static void main(String[] args) {
        long number = 0xF00000000000000FL;
        System.out.println("number:\n" + toBinaryString(number));

        long result = rotateLeft(number,2);
        System.out.println("number rotated left 2 bits:\n" +
        toBinaryString(result));

        result = rotateRight(number, 3);
        System.out.println("number rotated right 3 bits:\n" +
        toBinaryString(result));
    }
}

```



```

System.out.println(symbol);

if(symbol >= 'A'){
    if(symbol <= 'Z') {
        System.out.println("Capital: " + symbol);
    }
    else {
        if (symbol >= 'a'){
            if(symbol <= 'z'){
                System.out.println("Small: " + symbol);
            }
            else{
                System.out.println("Code is Greater but aint a letter");
            }
        }
    }
}
else {
    System.out.println("The Code is less than A and not a letter");
}

/*
=====
=====
Output:
'

The Code is less than A and not a letter

=====
===== */

/*
=====
=====
Output:
'

The Code is less than A and not a letter

=====
===== */

```

```
/*
=====
=====
Output:
```

h
Small: h

```
=====
===== */
=====
=====
*
* Roll No: 30
*
* File: 20.2-Deciphering_chars-2.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
=====
===== */
===== */
```

```
class LetterCheck2 {
    public static void main(String[] args) {
        char symbol = 'A';
        symbol = (char)(128.0*Math.random());
        System.out.println(symbol);

        if(symbol >= 'A' && symbol <= 'Z')
            System.out.println("You have the capital letter " + symbol);

        else {
            if(symbol >= 'a' && symbol <= 'z')
                System.out.println("You have the small letter " + symbol);
            else
                System.out.println("The code is not a letter");
        }
    }
}
```

```
/*
=====
=====
Output:
```

R

You have the capital letter R

```
=====
===== */
```

```
/*
=====
=====
```

Output:

#

The code is not a letter

```
=====
===== */
```

```
/*
=====
```

```
=====
*
```

```
=====
* Roll No: 30
```

```
=====
*
```

```
=====
* File: 20.3-Deciphering_chars-3.java
```

```
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
```

```
=====
*
```

```
=====
*
```

```
=====
===== */
```

```
import static java.lang.Character.isLowerCase;
import static java.lang.Character.isUpperCase;
```

```
class LetterCheck3 {
```

```
    public static void main(String[] args) {
```

```
        char symbol;
```

```
        symbol = (char)(128.0*Math.random());
```

```
        System.out.println(symbol);
```

```
        if(isUpperCase(symbol))
```

```
            System.out.println("You have the capital letter: " + symbol);
```

```
        else {
```

```
            if(isLowerCase(symbol))
```

```
                System.out.println("You have the small letter: " + symbol);
```

```
            else
```

```
        System.out.println("The code is not a letter");
    }
}
```

```
/*
=====
=====
Output:
```

|
The code is not a letter

```
=====
===== */ /*
=====
=====
=====
*
* Roll No: 30
*
* File: 21-ternary-or-conditional.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
===== */
```

```
class CheckEvenOdd {  
    public static void main( String args[] ) {  
        int number = 3;  
  
        String msg = (number % 2 == 0) ? "Even" : "Odd";  
        System.out.println(msg);  
    }  
}
```

```
}

/*
=====
=====

Output:
```

Odd!

```
=====
===== */ /*
=====
=====
=====
/*
* Roll No: 30
*
* File: 22-SwitchCase-with-Enum.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinrathod)
*
*/
=====
===== */

class SwitchUsingEnum{
    enum WashChoice {cotton, linen, wool, synthetic}

    public static void main(String[] args){
        WashChoice wash = WashChoice.cotton;

        // 1. shirts, 2. sweaters, 3. socks, 4. sheets, 5. pants
        int clothes = 3;

        switch(clothes){
            case 1:
                System.out.println("Washing Shirts");
                wash = WashChoice.cotton;
                break;

            case 2:
                System.out.println("Washing Sweaters");
                wash = WashChoice.wool;
                break;

            case 3:
                System.out.println("Washing socks");
                wash = WashChoice.wool;
                break;
            case 4:
                System.out.println("Washing Sheets");
                wash = WashChoice.linen;
                break;
            case 5:
                System.out.println("Washing Pants");
                wash = WashChoice.synthetic;
                break;
        }
    }
}
```

```

    default:
        System.out.println("Invalid Choice");
    }
}

}

```

/*
=====
=====
Output:

Washing socks

```

=====
=====
 */ /*
=====
=====
*
* Roll No: 30
*
* File: 23-ForLoop.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
=====
*/
```

```

class ForLoop{
    public static void main(String[] arg) {
        int[] numbers = { 10, 20, 30, 40, 50 };

        for (int i = 0; i < 5; i++){
            System.out.println(numbers[i]);
        }
    }
}
```

/*
=====
=====
Output:

```
10
20
30
40
50

=====
===== */ /*
=====
=====
*
* Roll No: 30
*
* File: 24-ForEach.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
===== */
===== */

class For_Each {
    public static void main(String[] arg) {
        int[] numbers = { 10, 20, 30, 40, 50 };

        for (int num : numbers)
            System.out.println(num);
    }
}

/*
=====
=====
Output:

10
20
30
40
50

=====
===== */ /*
=====
=====
```

```

/*
 * Roll No: 30
 *
 * File: 25-WhileLoop.java
 * Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
 *
 *
=====
===== */

```

```

class whileLoop {
    public static void main(String args[])
    {
        int i = 0;
        while (i++ < 3)
            System.out.println("Good morning Mam");
    }
}

```

```

/*
=====
=====
Output:

```

```

Good morning Mam
Good morning Mam
Good morning Mam

```

```

=====
===== */
=====
=====
*
* Roll No: 30
*
* File: 26-doWhileLoop.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
===== */

```

```

class DoWhileLoop {
    public static void main(String args[])
}

```

```

{
int i = 0;
do {
    System.out.println("Good morning Mam");
}while (i++ < 3);

}
}

```

```

/*
=====
=====
Output:

```

```

Good morning Mam
Good morning Mam
Good morning Mam
Good morning Mam

```

```

=====
=====
*/
/*
=====
=====
*
* Roll No: 30
*
* File: 27-Factorial.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
=====
*/

```

```

class Factorial{
    public static void main(String args){

        int i, ans = 1;
        int number = 5;

        for(i = 1;i <= number; i++)
            ans = ans * i;

        System.out.println("Factorial of " + number + " is: " + ans);
    }
}

```

```
}
```

```
/*
=====
=====
Output:
```

Factorial of 5 is: 120

```
=====
===== */
=====
=====
*
* Roll No: 30
*
* File: 28-LabelledContinue.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
=====
===== */

class Factorial2 {
    public static void main(String[] args) {
        long limit = 20L;
        long factorial = 1L;

        OuterLoop:
        for(long i = 1L; i <= limit; i++){
            factorial = 1;
            for(long j = 2L; j <= i; j++){
                if(i > 10L && i % 2L == 1L)
                    continue OuterLoop;
                factorial *= j;
            }

            System.out.println(i + "!" + factorial);
        }
    }
}
```

```
/*
=====
=====
Output:
```

```
1! is 1
2! is 2
3! is 6
4! is 24
5! is 120
6! is 720
7! is 5040
8! is 40320
9! is 362880
10! is 3628800
12! is 479001600
14! is 87178291200
16! is 20922789888000
18! is 6402373705728000
20! is 2432902008176640000
```

```
=====
===== */
/*
=====
=====
*
```

```
* Roll No: 30
*
* File:    29-prime.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*/
=====
===== */

class Prime{
    public static void main(String[] args){
        long num = 67280421310721L;

        if(num > 2 && num%2 == 0){
            System.out.println(num + " is not prime");
            System.exit(0);
        }

        int sqrt = (int)Math.sqrt(num) + 1;
```

```

        for(int i = 3; i < sqrt; i+=2){
            if(num % i == 0){
                System.out.println(num + " is not prime");
                System.exit(0);
            }
        }
        System.out.println(num + " is prime");
    }
}

```

/*
=====
=====
Output:

67280421310721 is prime

```

=====
===== */ /*
=====
===== =====
=====
*
* Roll No: 30
*
* File: 30-Random-Capital-Without-Vowel.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
===== */

```

```

class RandomCapital{
    public static void main(String[] args){
        System.out.println("This code will generate 5 Random" +
                           "Capital Letters without any Vowels in it");
        int count = 5;

        for(int i = 0; i < 5;){
            char c = (char)(128.0 * Math.random());
            if(c > 65 && c <=90){
                i++;
                if (c == 'E' || c == 'I' || c == 'O' || c == 'U')
                    i--;
                else
                    System.out.print(c);
            }
        }
    }
}

```

```
        }
    }
}
```

```
/*
=====
=====
Output:
```

This code will generate 5 RandomCapital Letters without any Vowels in it
SGJCR

```
=====
*/

```

```
/*
=====
=====
Output:
```

This code will generate 5 RandomCapital Letters without any Vowels in it
VZBCG

```
=====
*/

```

```
/*
=====
=====
Output:
```

This code will generate 5 RandomCapital Letters without any Vowels in it
GJPRW

```
=====
*/

```

```
/*
=====
=====
Output:
```

This code will generate 5 RandomCapital Letters without any Vowels in it
HXCSR

```
=====
=====
*/
/*
=====
=====
=====
Output:
```

This code will generate 5 RandomCapital Letters without any Vowels in it
YXFDH

```
=====
=====
*/
 /**
=====
=====
=====
*
* Roll No: 30
*
* File: 31-Help.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*/
=====
=====
*/

```

```
class Help3 {
    public static void main(String args[])
        throws java.io.IOException {
        char choice, ignore;
        for(;;) {
            do {
                System.out.println("Help on:");
                System.out.println(" 1. if");
                System.out.println(" 2. switch");
                System.out.println(" 3. for");
                System.out.println(" 4. while");
                System.out.println(" 5. do-while");
                System.out.println(" 6. break");
                System.out.println(" 7. continue\n" );
                System.out.print ("Choose one (q to quit): ");
                choice = (char) System.in.read();
```

```

do {
    ignore = (char) System.in.read();
} while(ignore != '\n');
} while( choice < '1' | choice > '7' & choice != 'q');
if(choice != 'q') break;
System.out.println("\n");
switch(choice) {
    case '1':
        System.out.println("The if:\n");
        System.out.println ("if (condition) statement ;");
        System.out.println ("else statement ;");
        break;
    case '2':
        System.out.println ("The ewitch:\n");
        System.out.println("switch(expression) ()");
        System.out.println(" case constant :");
        System.out.println ("statement sequence");
        System.out.println (" break;");
        System.out.println(" // ...");
        System.out .println("}");
        break;
    case '3':
        System.out.println("The for:\n");
        System.out.print("for(init; condition; iteration)");
        System.out.println(" statement;");
        break;
    case '4':
        System.out.println ("The while: \n");
        System.out.println ("while(condition) statement;");
        break;
    case '5':
        System.out.println("The do-while:\n") ;
        System.out.println("do ()");
        System.out.println(" statement ;");
        System.out.println(") while (condition) 7");
        break;
    case '6':
        System.out.println("The break: \n");
        System.out.println("*break; or break label;");
        break;
    case '7':
        System.out.println("The continue: \n");
        System.out.println("continue; or continue label;");
        break;
}
System.out.println();
}

```

```

    }
}

/*
=====
=====
*
* Roll No: 30
*
* File:      32-roundingError.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinrzrathod)
*
*
===== */
===== */

// Show square roots of 1 to 99 and the rounding error.
class SqrRoot {
    public static void main(String args[]) {
        double num, sroot, rerr;

        for (num = 1.0; num < 100.0; num++) {

            sroot = Math.sqrt (num) ;
            System.out.println("Square root of " + num + " is " + sroot);

            rerr = num - (sroot * sroot);
            System.out.println("Rounding error is " + rerr + "\n");
        }
    }
}

/*
=====
=====
Output:

Square root of 1.0 is 1.0
Rounding error is 0.0

Square root of 2.0 is 1.4142135623730951
Rounding error is -4.440892098500626E-16

Square root of 3.0 is 1.7320508075688772
Rounding error is 4.440892098500626E-16

Square root of 4.0 is 2.0

```

Rounding error is 0.0

Square root of 5.0 is 2.23606797749979

Rounding error is -8.881784197001252E-16

Square root of 6.0 is 2.449489742783178

Rounding error is 8.881784197001252E-16

Square root of 7.0 is 2.6457513110645907

Rounding error is -8.881784197001252E-16

Square root of 8.0 is 2.8284271247461903

Rounding error is -1.7763568394002505E-15

Square root of 9.0 is 3.0

Rounding error is 0.0

Square root of 10.0 is 3.1622776601683795

Rounding error is -1.7763568394002505E-15

Square root of 11.0 is 3.3166247903554

Rounding error is 0.0

Square root of 12.0 is 3.4641016151377544

Rounding error is 1.7763568394002505E-15

Square root of 13.0 is 3.605551275463989

Rounding error is 1.7763568394002505E-15

Square root of 14.0 is 3.7416573867739413

Rounding error is 0.0

Square root of 15.0 is 3.872983346207417

Rounding error is -1.7763568394002505E-15

Square root of 16.0 is 4.0

Rounding error is 0.0

Square root of 17.0 is 4.123105625617661

Rounding error is 0.0

Square root of 18.0 is 4.242640687119285

Rounding error is 3.552713678800501E-15

Square root of 19.0 is 4.358898943540674

Rounding error is -3.552713678800501E-15

Square root of 20.0 is 4.47213595499958
Rounding error is -3.552713678800501E-15

Square root of 21.0 is 4.58257569495584
Rounding error is 0.0

Square root of 22.0 is 4.69041575982343
Rounding error is 0.0

Square root of 23.0 is 4.795831523312719
Rounding error is 3.552713678800501E-15

Square root of 24.0 is 4.898979485566356
Rounding error is 3.552713678800501E-15

Square root of 25.0 is 5.0
Rounding error is 0.0

Square root of 26.0 is 5.0990195135927845
Rounding error is 3.552713678800501E-15

Square root of 27.0 is 5.196152422706632
Rounding error is 0.0

Square root of 28.0 is 5.291502622129181
Rounding error is -3.552713678800501E-15

Square root of 29.0 is 5.385164807134504
Rounding error is 3.552713678800501E-15

Square root of 30.0 is 5.477225575051661
Rounding error is 0.0

Square root of 31.0 is 5.5677643628300215
Rounding error is 3.552713678800501E-15

Square root of 32.0 is 5.656854249492381
Rounding error is -7.105427357601002E-15

Square root of 33.0 is 5.744562646538029
Rounding error is 0.0

Square root of 34.0 is 5.830951894845301
Rounding error is 0.0

Square root of 35.0 is 5.916079783099616
Rounding error is 0.0

Square root of 36.0 is 6.0
Rounding error is 0.0

Square root of 37.0 is 6.082762530298219
Rounding error is 7.105427357601002E-15

Square root of 38.0 is 6.164414002968976
Rounding error is 7.105427357601002E-15

Square root of 39.0 is 6.244997998398398
Rounding error is 0.0

Square root of 40.0 is 6.324555320336759
Rounding error is -7.105427357601002E-15

Square root of 41.0 is 6.4031242374328485
Rounding error is 0.0

Square root of 42.0 is 6.48074069840786
Rounding error is 0.0

Square root of 43.0 is 6.557438524302
Rounding error is 7.105427357601002E-15

Square root of 44.0 is 6.6332495807108
Rounding error is 0.0

Square root of 45.0 is 6.708203932499369
Rounding error is -7.105427357601002E-15

Square root of 46.0 is 6.782329983125268
Rounding error is 0.0

Square root of 47.0 is 6.855654600401044
Rounding error is 0.0

Square root of 48.0 is 6.928203230275509
Rounding error is 7.105427357601002E-15

Square root of 49.0 is 7.0
Rounding error is 0.0

Square root of 50.0 is 7.0710678118654755
Rounding error is -7.105427357601002E-15

Square root of 51.0 is 7.14142842854285

Square root of 51.0 is 7.105427357601002E-15
Rounding error is -7.105427357601002E-15

Square root of 52.0 is 7.211102550927978
Rounding error is 7.105427357601002E-15

Square root of 53.0 is 7.280109889280518
Rounding error is 0.0

Square root of 54.0 is 7.3484692283495345
Rounding error is 0.0

Square root of 55.0 is 7.416198487095663
Rounding error is 0.0

Square root of 56.0 is 7.483314773547883
Rounding error is 0.0

Square root of 57.0 is 7.54983443527075
Rounding error is 0.0

Square root of 58.0 is 7.615773105863909
Rounding error is -7.105427357601002E-15

Square root of 59.0 is 7.681145747868608
Rounding error is 7.105427357601002E-15

Square root of 60.0 is 7.745966692414834
Rounding error is -7.105427357601002E-15

Square root of 61.0 is 7.810249675906654
Rounding error is 7.105427357601002E-15

Square root of 62.0 is 7.874007874011811
Rounding error is 0.0

Square root of 63.0 is 7.937253933193772
Rounding error is -7.105427357601002E-15

Square root of 64.0 is 8.0
Rounding error is 0.0

Square root of 65.0 is 8.06225774829855
Rounding error is 1.4210854715202004E-14

Square root of 66.0 is 8.12403840463596
Rounding error is -1.4210854715202004E-14

Square root of 67.0 is 8.18535277187245

Rounding error is 0.0

Square root of 68.0 is 8.246211251235321

Rounding error is 0.0

Square root of 69.0 is 8.306623862918075

Rounding error is 0.0

Square root of 70.0 is 8.366600265340756

Rounding error is 0.0

Square root of 71.0 is 8.426149773176359

Rounding error is 0.0

Square root of 72.0 is 8.48528137423857

Rounding error is 1.4210854715202004E-14

Square root of 73.0 is 8.54400374531753

Rounding error is 1.4210854715202004E-14

Square root of 74.0 is 8.602325267042627

Rounding error is 0.0

Square root of 75.0 is 8.660254037844387

Rounding error is -1.4210854715202004E-14

Square root of 76.0 is 8.717797887081348

Rounding error is -1.4210854715202004E-14

Square root of 77.0 is 8.774964387392123

Rounding error is -1.4210854715202004E-14

Square root of 78.0 is 8.831760866327848

Rounding error is -1.4210854715202004E-14

Square root of 79.0 is 8.888194417315589

Rounding error is 0.0

Square root of 80.0 is 8.94427190999916

Rounding error is -1.4210854715202004E-14

Square root of 81.0 is 9.0

Rounding error is 0.0

Square root of 82.0 is 9.055385138137417

Rounding error is -1.4210854715202004E-14

Square root of 83.0 is 9.1104335791443
Rounding error is 0.0

Square root of 84.0 is 9.16515138991168
Rounding error is 0.0

Square root of 85.0 is 9.219544457292887
Rounding error is 0.0

Square root of 86.0 is 9.273618495495704
Rounding error is 0.0

Square root of 87.0 is 9.327379053088816
Rounding error is -1.4210854715202004E-14

Square root of 88.0 is 9.38083151964686
Rounding error is 0.0

Square root of 89.0 is 9.433981132056603
Rounding error is 1.4210854715202004E-14

Square root of 90.0 is 9.486832980505138
Rounding error is 0.0

Square root of 91.0 is 9.539392014169456
Rounding error is 0.0

Square root of 92.0 is 9.591663046625438
Rounding error is 1.4210854715202004E-14

Square root of 93.0 is 9.643650760992955
Rounding error is 0.0

Square root of 94.0 is 9.695359714832659
Rounding error is -1.4210854715202004E-14

Square root of 95.0 is 9.746794344808963
Rounding error is 1.4210854715202004E-14

Square root of 96.0 is 9.797958971132712
Rounding error is 1.4210854715202004E-14

Square root of 97.0 is 9.848857801796104
Rounding error is 1.4210854715202004E-14

Square root of 98.0 is 9.899494936611665

Rounding error is 0.0

Square root of 99.0 is 9.9498743710662

Rounding error is 0.0

```
=====
===== */ /*
=====
=====
=====
*
* Roll No: 30
*
* File: 33-loopUntilKeypress.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
===== */
===== */

class ForTest {
    public static void main(String args[]) throws java.io.IOException {
        int i;
        System.out.println("Press small \"a\" to stop.");
        for(i = 0; (char) System.in.read() != 'a'; i++)
            System.out.println("\"a\" not pressed. Press \"a\" ");
    }
}

/*
=====
=====
Press small "a" to stop.
q
"a" not pressed. Press "a"
"a" not pressed. Press "a"
w
"a" not pressed. Press "a"
"a" not pressed. Press "a"
a

*
=====
===== */
===== */
```

```

/*
=====
=====
*
* Roll No: 30
*
* File:    34-WhileLoop.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
===== */

```

```

class WhileDemo {
    public static void main(String args[]) {
        char ch;
        ch = 'a';

        while(ch <= 'z') {
            System.out.print (ch) ;
            ch++;
        }
    }
}

```

```

/*
=====
=====
Output:

```

abcdefghijklmnopqrstuvwxyz

```

=====
===== */
=====
=====
*
* Roll No: 30
*
* File:    35-DWLoop.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
===== */

```

```

// Demonstrate the do-while loop.
class DWDemo {
    public static void main(String args[])
        throws java.io.IOException {
        char ch;
        do {
            System.out.print ("Press a key followed by ENTER: ");
            ch = (char) System.in.read();
        } while(ch != 'q');
    }
}

/*
=====
=====
* Output:
Press a key followed by ENTER: qw

*
=====
=====
*/
/*
=====
=====
* Roll No: 30
*
* File: 36-GuessLetter.java
* Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
*
*
=====
=====
*/
class GuessLetter{
    public static void main(String[] args){
        char c = args[0].charAt(0);
        int count = 1;

        for(;;){
            char symbol = (char)(128.0 * Math.random());
            if(symbol == c){
                System.out.print("Correct Guess ");
                System.out.println("at " + count + " time");
                break;
            }
        }
    }
}

```

```
        else{
            System.out.println("Oops! The Letter was: " + symbol);
            count++;
        }
    }
}
```

```
/*
=====
=====
Output:
```

GuessLetter w

```
Oops! The Letter was: e
Oops! The Letter was: N
Oops! The Letter was: p
Oops! The Letter was: )
Oops! The Letter was: l
Oops! The Letter was: #
Oops! The Letter was: !
Oops! The Letter was:
Oops! The Letter was: 0
Oops! The Letter was: E
Oops! The Letter was: A
Oops! The Letter was: )
Oops! The Letter was: #
Oops! The Letter was: x
Oops! The Letter was: !
Oops! The Letter was: @
Oops! The Letter was: 3
Oops! The Letter was: x
Oops! The Letter was: #
Oops! The Letter was: `
Oops! The Letter was: #
Oops! The Letter was: +
Oops! The Letter was: :
Oops! The Letter was: f
Oops! The Letter was: .
Oops! The Letter was: N
Oops! The Letter was: F
Oops! The Letter was: h
Oops! The Letter was: o
Oops! The Letter was: #
Oops! The Letter was:
```

Oops! The Letter was: #
 Oops! The Letter was: -
 Oops! The Letter was: "
 Oops! The Letter was: 6
 Oops! The Letter was: =
 Oops! The Letter was: 6
 Oops! The Letter was: h
 Oops! The Letter was: #
 Oops! The Letter was: J
 Oops! The Letter was: x
 Oops! The Letter was: d
 Oops! The Letter was:
 Oops! The Letter was: 6
 Oops! The Letter was: j
 Oops! The Letter was:
 Oops! The Letter was: q
 Oops! The Letter was: #
 Oops! The Letter was: >
 Oops! The Letter was: 6
 Oops! The Letter was: Y
 Oops! The Letter was: P
 Correct Guess at 53 time

```

=====
===== */
/*
=====
=====
=====
 *
 * Roll No: 30
 *
 * File:      37-ToggleCase.java
 * Copyright: 17-Sep-2020 by Ajinkya Rathod(ajinzrathod)
 *
=====
===== */
===== */

class ToggleCase{
  public static void main(String[] args){

    // Checking if only 1 argument is passed
    if(args.length > 1 || args.length < 1){
      System.out.println("Enter only 1 string in arg");
      System.exit(1);
  }
}

```

```
// Reading 1st argument
String str1 = args[0];

// Toggling Case
for(int i = 0; i < str1.length(); i++){
    char c = str1.charAt(i);

    // Capital to Small
    if(c >= 65 && c<=90)
        c += 32;

    // Small to Capital
    else if(c >= 97 && c <= 122)
        c -= 32;

    System.out.print(c);
}
System.out.println("");
}

/*
=====
=====
Output:
ToggleCase AjInKyA
aJiNkYa
=====
*/
```

||||||||||||||||||||||||||||||||||||||||

Assertion

```
=====
=====
NAME : Ajinkya Rathod
ROLL NO : 30
CLASS : MCA 3
SUBJECT : JAVA
=====
```

ASSIGNMENT - 3

```
=====
=====
Q1: Assertion 1
=====
=====
class Assertion1 {
    public static void main(String args[]) throws java.io.IOException
    {
        int value=18;

        assert value>=20 : "Eligible";
        System.out.println("Value:" + value);
    }
}
```

OUTPUT:

```
java Assertion1
Value:18
=====
=====
Q2: Assertion 2
=====
=====
import java.util.Scanner;

class Assertion2{
    public static void main(String args[]) throws java.io.IOException
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Your Age ");
        int value = scanner.nextInt();
```

```

        assert value>=18 : "Not Valid";
        System.out.println("Value:" + value);
    }
}
=====
```

OUTPUT:

```

java Assertion2
Enter Your Age 21
Value:21
=====
```

Q3:Example of Scanner..

```

import java.util.Scanner;
class Scanner_test{
    public static void main(String args[]){
        Scanner scn = new Scanner(System.in);
        System.out.print("Enter your name, age, salary :");

        String name = scn.nextLine();

        int age = scn.nextInt();

        double salary = scn.nextDouble();

        System.out.println("Name = "+name);
        System.out.println("Age = "+age);
        System.out.println("Salary = "+salary);
    }
}
=====
```

OUTPUT:

```

java Scanner_test
Enter your name, age, salary :Ajinkya
21
30000
Name = Ajinkya
Age = 21
Salary = 30000.0
=====
```

Q4:Systems properties..

```

import java.util.*;

class TryPro {
    public static void main(String[] args) {
```

```
Properties properties = System.getProperties();
properties.list(System.out);
}
=====
=====
```

OUTPUT:

```
java TryPro
-- listing properties --
java.specification.version=15
sun.management.compiler=HotSpot 64-Bit Tiered Compilers
sun.jnu.encoding=UTF-8
java.runtime.version=15.0.1+9-18
java.class.path=.
user.name=ajinzrathod
java.vm.vendor=Oracle Corporation
path.separator=:
sun.arch.data.model=64
os.version=5.4.0-52-generic
java.runtime.name=Java(TM) SE Runtime Environment
file.encoding=UTF-8
java.vendor.url=https://java.oracle.com/
java.vm.name=Java HotSpot(TM) 64-Bit Server VM
java.vm.specification.version=15
os.name=Linux
sun.java.launcher=SUN_STANDARD
user.country=IN
sun.boot.library.path=/usr/lib/jvm/java-15-oracle/lib
sun.java.command=TryPro
java.vendor.url.bug=https://bugreport.java.com/bugreport/
java.io.tmpdir=/tmp
jdk.debug=release
sun.cpu.endian=little
java.version=15.0.1
user.home=/home/ajinzrathod
user.dir=/home/ajinzrathod/Desktop
os.arch=amd64
user.language=en
java.specification.vendor=Oracle Corporation
java.vm.specification.name=Java Virtual Machine Specification
java.version.date=2020-10-20
java.home=/usr/lib/jvm/java-15-oracle
file.separator=/
java.vm.compressedOopsMode=32-bit
line.separator=

java.library.path=/usr/java/packages/lib:/usr/lib64:/li...
java.vm.info=mixed mode, sharing
java.vm.specification.vendor=Oracle Corporation
java.specification.name=Java Platform API Specification
java.vendor=Oracle Corporation
```

```

java.vm.version=15.0.1+9-18
sun.io.unicode.encoding=UTF-8
java.class.version=59.0
=====
=====
Q5:2-d Array.
=====
=====

class twoDArray{
    public static void main(String args[]){
        int arr[][] = {{2,7,9},{3,6,1},{7,4,2}};

        for(int i = 0; i < 3; i++)
        {
            for(int j = 0; j < 3; j++)
            {
                System.out.print(arr[i][j] + " ");
            }
            System.out.println();
        }
    }
}
=====

=====


```

OUTPUT:

```

java twoDArray
2 7 9
3 6 1
7 4 2
=====

=====
```

Q6:3-d Array..

```

=====
=====

class threeDArray{
    public static void main(String args[]){
        int arr[][][] = {{{

            {2,7,9},
            {3,6,1},
            {7,4,2} },
            {

            {1,2,3},
            {4,5,6},
            {7,8,9} },
            };

        for(int i = 0; i < 2; i++)
        {
            for(int j = 0; j < 3; j++)
            {
                for(int k = 0; k < 3; k++)

```

```

        {
            System.out.print(arr[i][j][k] + " ");
        }
    }
}
=====
=====
```

OUTPUT:

```
java threeDArray
2 7 9 3 6 1 7 4 2
1 2 3 4 5 6 7 8 9
```

```
=====
=====
```

Q7:JAGGED Array.

```
=====
=====
```

```
import java.util.*;

class jaggedArray {
    public static void main(String[] args) {
        int arr[][] = new int[2][];
        //first row has three columns
        arr[0]= new int[3];
        //secound row has 2 columns
        arr[1] =new int[2];
        int count =0;
        for(int i=0;i<arr.length; i++)
        {
            for(int j=0;j<(arr[i].length); j++)
            {
                arr[i][j] = count++;
            }
        }
        System.out.println("Contents of 2D jagged Array");
        for(int i=0;i<arr.length;i++)
        {
            for(int j=0;j<arr[i].length;j++)
                System.out.print(arr[i][j] + " ");
            System.out.println();
        }
    }
}
```

```
=====
=====
OUTPUT:
```

```
java jaggedArray
Contents of 2D jagged Array
0 1 2
3 4
=====
=====
Q8:Array1 example..
=====
=====
import java.util.*;

class Array1 {
    public static void main(String[] args) {
        int arr[] = new int [5];

        for(int i=5;i>0;i--)
            arr[5-i] = i;

        Arrays.sort(arr);

        for(int i=0;i<5;++)
            System.out.print(arr[i]);
    }
}
```

```
=====
=====
```

OUTPUT:

```
java Array1
12345
=====
=====
Q9:String Collection./ count the number of vowels..
=====
=====
import java.util.*;

class StringCollection {
    public static void main(String[] args) {

        String phrase = "The Quick";
        int vowels = 0;

        for(char ch : phrase.toCharArray())
        {
            ch = Character.toLowerCase(ch);
            if( ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
```

```

    {
        ++vowels;

    }
}

=====
=====

System.out.println("The Phrase contains " + vowels + " vowels.");
}
=====
```

OUTPUT:

```
java StringCollection
The Phrase contains 3 vowels.
```

```
=====
=====

Q11: Test String.
=====
```

```
import java.util.*;

class TestString {
    public static void main(String[] args) {

        String s1 = "Test";
        String s2 = "Test";
        String s3 = new String("Test");
        final String s4 = s3.intern();
        System.out.println(s1==s2);
        System.out.println(s2==s3);
        System.out.println(s3==s4);
        System.out.println(s1==s3);
        System.out.println(s1==s4);
        System.out.println(s1.equals(s2));
        System.out.println(s2.equals(s3));
        System.out.println(s3.equals(s4));
        System.out.println(s1.equals(s4));
        System.out.println(s1.equals(s3));

    }
}
```

OUTPUT:

```
java TestString
true
false
false
```

```
false
true
true
true
true
true
true

=====
=====
Q12:Lucky Stars (pg. 156 - Ivor horton)
=====

import java.util.*;

class LuckyStars {
    public static void main(String[] args) {

        String[] stars = {
            "Robert Redford" , "Marilyn Monroe", "Boris Karloff" , "Lassie",
            "Hopalong Cassidy", "Trigger"
        };

        System.out.println("Your Luckey Stars for today is " + stars[(int)
(stars.length * Math.random())]);
    }
}
=====
```

OUTPUT:

```
java LuckyStars
Your Luckey Stars for today is Boris Karloff
```

```
=====
=====
Q13:Try It Out : String Concatenation page no : 189/158
=====

public class joinstring {
    public static void main(String[] args) {
        String firststring = "Many" ;
        String secondstring = "hands" ;
        String thirdstring = "make light work" ;

        String mystring ; //variable to store results

        //join three strings and store the results
        mystring = firststring + " " +secondstring + " " + thirdstring ;
        System.out.println(mystring);
```

```

//convert an integer to string and join with two other string
int num = 99 ;
mystring = num + " " + secondstring + " "+ thirdstring ;
System.out.println(mystring);

//combining a string and integers

mystring = "fifty five is " + 5 + 5 ;
System.out.println(mystring);

//combining a integer and string

mystring = 5 + 5 + " is ten ";
System.out.println(mystring);
}

}

=====
=====

OUTPUT:

javac joinstring.java

java joinstring
Many hands make light work
99 hands make light work
fifty five is 55
10 is ten

=====

=====

Q14:Try It Out : String Identity page no : 194 / 163
=====

=====

public class matchstring {
    public static void main(String[] args) {
        String string1 = "Too many";
        String string2 = " cooks";
        String string3 = "Too many cooks";

        string1 += string2 ;

        //Doisplay the concates of the strings

        System.out.println("\nTest 1 ");
        System.out.println("String 3 is now : " + string3);
        System.out.println("String 1 is now : " + string1);

        //Testing the equality
    }
}

```

```

if(string1.equals(string3)){
    System.out.println("\nString 1 and string 3 is equal ");
}
else{
    System.out.println("\nString 1 and string 3 is not equal ");
}

//changing the case and checking the methods are equal or not
string3 = "TOO many cooks" ;

System.out.println("\nTest 2 ");
System.out.println("String 3 is now : " + string3);
System.out.println("String 1 is now : " + string1);

if(string1.equals(string3)){
    System.out.println("\nString 1 and string 3 is equal comparing the case");
}
else{
    System.out.println("\nString 1 and string 3 is not equal comparing the
case");
}

if(string1.equalsIgnoreCase(string3)){
    System.out.println("\nString 1 and string 3 is equal comparing the case");
}
else{
    System.out.println("\nString 1 and string 3 is not equal comparing the
case");
}
}
=====
```

OUTPUT:

javac matchstring.java

java matchstring

Test 1

String 3 is now : Too many cooks
String 1 is now : Too many cooks

String 1 and string 3 is equal

Test 2

String 3 is now : TOO many cooks
String 1 is now : Too many cooks

SString 1 and string 3 is not equal comparing the case

SString 1 and string 3 is equal comparing the case

```
=====
=====
Q16: Some string operation from java a book page no 184
=====
```

```
public class strops {
    public static void main(String[] args) {
        String str1 = "When it comes to web programming , Java is #1.";
        String str2 = new String(str1);
        String str3 = "Java strings are powerful.";

        int result,idx;
        char ch;
        System.out.println("Length of str1 : " + str1.length());

        //display str1, one char at a time

        for(int i = 0 ; i < str1.length() ; i++){
            System.out.print(str1.charAt(i));
        }
        System.out.println();

        if(str1.equals(str2)){
            System.out.println("str1 equals str2 ");
        }
        else{
            System.out.println("str1 does not equals str2");
        }

        if(str1.equals(str3)){
            System.out.println("str1 equals str3 ");
        }
        else{
            System.out.println("str1 does not equals str3");
        }

        //compare two strings

        result = str1.compareTo(str3) ;

        if(result == 0){
            System.out.println("str1 and str3 are equals ");
        }
        else if(result < 0 ){
            System.out.println("str1 is less than str3 ");
        }
        else{
```

```

        System.out.println("str1 is greater than str3");
    }

    //assigning a new string t str2

    str2 = "One Two Three One";

    // returning the index of the given word

    idx = str2.indexOf("One");
    System.out.println("Index of the first occurence of one : "+ idx);

    idx = str2.lastIndexOf("One");
    System.out.println("Index of the last occurence of one : "+ idx);

}
}

=====
=====
```

OUTPUT:

```

java strops
Length of str1 : 46
When it comes to web programming , Java is #1.
str1 equals str2
str1 does not equals str3
str1 is greater than str3
Index of the first occurence of one : 0
Index of the last occurence of one : 14
```

```

=====
=====
```

Q17: Arrays of Strings page no 185 of java A

```

=====
=====
```

```

public class stringarray {
    public static void main(String[] args) {
        String str[] = { "This" , "is" , "a" , "test ."};
        System.out.println("Original Array : ");

        for(String s : str){
            System.out.print(s + " ");
        }

        System.out.println("\n");
        //change the string
        str[1] = "was";
    }
}
```

```

        str[3] = "test, too!";

        System.out.println("Modified Array :");

        for(String s : str){
            System.out.print(s + " ");
        }
    }
}
=====
```

OUTPUT:

```

java stringarray
Original Array :
This is a test .
```

```

Modified Array :
This was a test, too!
```

Q18:Use substring from java A page no 187

```

public class substr {
    public static void main(String[] args) {
        String orgstr = "Java makes the Web move";

        // construct a substring

        String substr = orgstr.substring(5,18);

        System.out.println("Orgstr : " + orgstr);
        System.out.println("substr : " + substr);

    }
}
```

OUTPUT:

```
javac substr.java
```

```

java substr
Orgstr : Java makes the Web move
substr : makes the Web
```

```
=====
=====
Q19: Use a string to control a switch statement from java A book page no 188
=====
=====

public class stringswitch {
    public static void main(String[] args) {
        String command = "cancel";

        switch(command){
            case "connect":
                System.out.println("connecting");
                break;
            case "cancel":
                System.out.println("canceling");
                break;
            case "disconnect":
                System.out.println("disconnecting");
                break;
            default:
                System.out.println("Command Error!");
                break;
        }
    }
}
```

=====

=====

=====

OUTPUT:

```
java stringswitch
canceling
```

```
=====
=====
Q20: Using a StringBuffer Object to Assemble a String  from ivor horton page
226
=====
=====
```

```
public class userstringbuffer {
    public static void main(String[] args) {
        StringBuffer sentence = new StringBuffer(20);

        System.out.println("\nString buffer objects capacity is " +
sentence.capacity() + " and string length is : " + sentence.length());

        //Append all the words to the StringBuffer object

        String[] words = {"too" , "many" , "cooks" , "spoil" , "the" , "broth"};
```

```

sentence.append(words[0]);

for(int i = 1; i < words.length ;i++){
    sentence.append(' ').append(words[i]);
}

//Show the result

System.out.println("\n stringin string buffer object is : \n" +
sentence.toString());

System.out.println("\nString buffer objects capacity is " +
sentence.capacity() + " and string length is : " + sentence.length());

sentence.insert(sentence.lastIndexOf("cooks")+4,"ie");
sentence.insert(sentence.lastIndexOf("broth")+5,"er");

System.out.println("\n stringin string buffer object is : \n" + sentence);

System.out.println("\nString buffer objects capacity is " +
sentence.capacity() + " and string length is : " + sentence.length());

}

=====
=====

OUTPUT:
```

java userstringbuffer

String buffer objects capacity is 20 and string length is : 0

stringin string buffer object is :
too many cooks spoil the broth

String buffer objects capacity is 42 and string length is : 30

stringin string buffer object is :
too many cookies spoil the brother

String buffer objects capacity is 42 and string length is : 34

```
=====
=====

Q21:Exciting Concordance Entries (pg. 205-Ivor Horton)
=====
```

```

public class excorentri
{
    public static void main(String[] args)
```

```

{
    // Text string to be analyzed
    String text = "To be or not to be, that is the question;" +
        " Whether 'tis nobler in the mind to suffer" +
        " the slings and arrows of outrageous fortune," +
        " or to take arms against a sea of troubles," +
        " and by opposing end them?";

    int andCount = 0;
    // Number of and's
    int theCount = 0;
    // Number of the's
    int index = -1;
    // Current index position
    String andStr = "and";
    // Search substring
    String theStr = "the";
    // Search substring
    // Search forwards for "and"
    index = text.indexOf(andStr);
    // Find first 'and'
    while(index >= 0)
    {
        ++andCount;
        index += andStr.length();
        // Step to position after last 'and'
        index = text.indexOf(andStr, index);
    }
    // Search backwards for "the"
    index = text.lastIndexOf(theStr);
    // Find last 'the'
    while(index >= 0)
    {
        ++theCount;
        index -= theStr.length();
        // Step to position before last 'the'
        index = text.lastIndexOf(theStr, index);
    }
    System.out.println("The text contains " + andCount + " and's\n" + "The
text contains " + theCount + " the's");
}
=====
=====

OUTPUT :
java excorentri.java
The text contains 2 and's
The text contains 5 the's
=====
=====
Q22:StringTokenizing (pg. 211 - Ivor Horton)

```

```
=====
=====

class StringTokenizer
{
    public static void main(String[] args)
    {
        String text = "To be or not to be, that is the question." // String to
segment
        String delimiters = "[, .]";
        // Delimiters are comma, space, and period
        int[] limits = {0, -1};
        // Limit values to try
        // Analyze the string
        for(int limit : limits)
        {
            System.out.println("\nAnalysis with limit = " + limit);
            String[] tokens = text.split(delimiters, limit);
            System.out.println("Number of tokens: " + tokens.length);
            for(String token : tokens)
            {
                System.out.println(token);
            }
        }
    }
}
```

```
=====
=====

OUTPUT :
java StrToken.java
```

Analysis with limit = 0
Number of tokens: 11
To
be
or
not
to
be

that
is
the
question

Analysis with limit = -1
Number of tokens: 12
To
be
or

```

not
to
be

that
is
the
question
=====
=====

Q23:Exercise 1 to 5 (pg. 229 - Ivor Horton)
=====

=====

Exercise 1:
=====
public class MonthAndAverage {
    public static void main(String args[]) {
        String[] monthNames = {
            "January" , "February", "March" , "April",
            "May" , "June" , "July" , "August",
            "September","October" , "November", "December"
        };

        double average = 0.0;
        double[] numbers = new double[12];

        for(int i = 0 ; i<numbers.length ; i++) {
            numbers[i] = Math.random()*100.0;
            System.out.println(monthNames[i] + " Generated " + numbers[i]);
            average += numbers[i];
        }
        average /= numbers.length;

        System.out.println("\nAverage of Random Generated numbers is " +
average);
    }
}

/
=====

=====
Output :
java MonthAndAverage.java

January Generated 27.141041721717663
February Generated 67.87611296787547
March Generated 83.45406765362513
April Generated 27.909407920488473
May Generated 17.74974470039815
June Generated 36.54939032872846
July Generated 75.57601375964221
August Generated 39.22331224741854

```

September Generated 83.73890273217255
 October Generated 23.70713053346526
 November Generated 48.62216087397412
 December Generated 52.693662098405866

Average of Random Generated numbers is 48.68674562815932

=====

Exercise 2:

=====**

```
public class Sort {
    public static void main(String args[]) {

        String text = "A method contains one or more statements Methods are
subroutines that manipulate the data "+

            "defined by the class and, in many cases,provide access to that data";
        int count = 0;
        boolean isWord = false;
        for (int i = 0 ; i<text.length() ; i++) {
            if(isWord) {
                if(Character.isLetter(text.charAt(i)) || text.charAt(i) == '\\')
                    continue;
                else
                    isWord = false;
            }
            else if(Character.isLetter(text.charAt(i))) {

                count++;
                isWord = true;
            }
        }
    }
}
```

String[] words = new String[count];

```
int start = 0;
int wordIndex = 0;
isWord = false;
for (int i = 0 ; i<text.length() ; i++) {
    if(!isWord) {
        if(Character.isLetter(text.charAt(i))) {
            start = i;
            isWord = true;
        }
    }
    else {
        if(Character.isLetter(text.charAt(i)) || text.charAt(i) == '\\')
            continue;
        else {
```

```

        isWord = false;
        words[wordIndex++] = text.substring(start,i);
    }
}
}

if(wordIndex < words.length)
    words[wordIndex] = text.substring(start);

String temp = null;
boolean exchange = true;
while(exchange) {
    exchange = false;
    for(int i = 1 ; i<words.length ; i++) {
        if(words[i-1].compareTo(words[i])>0) {
            temp = words[i];
            words[i] = words[i-1];
            words[i-1] = temp;
            exchange = true;
        }
    }
}

for(String word : words) {
    System.out.println(word);
}

}

/
=====
=====
Output :

```

java MonthAndAverage.java

A
Methods
access
and
are
by
cases
class
contains
data
data
defined
in
manipulate

```

many
method
more
one
or
provide
statements
subroutines
that
that
the
the
to
=====
```

Exercise 3:

```
=====**=====
```

Exercise 5:

```
=====**=====
```

```
public class Reverse {
```

```
    public static void main(String args[]) {
```

```
        String text = "A method contains one or more statements Methods are
subroutines that manipulate the data "+
```

```
            "defined by the class and, in many cases, provide access to that data";
        boolean isWord = false;
        int start = 0;
        StringBuffer reversedText = new StringBuffer();
        StringBuffer word = new StringBuffer();
        for(int i = 0 ; i<text.length() ; i++) {
            if(!isWord) {
                if(Character.isLetter(text.charAt(i))) {
                    word.append(text.charAt(i));
                    isWord = true;
                } else {
                    reversedText.append(text.charAt(i));
                }
            } else {
                if(Character.isLetter(text.charAt(i)) || text.charAt(i) == '\'') {
                    word.append(text.charAt(i));
                    continue;
                }
                else {
                    reversedText.append(word.reverse());
                    reversedText.append(text.charAt(i));
                    word.delete(0,word.length());
                    isWord = false;
                }
            }
        }
    }
```

```

        }
    }
    if(word.length()>0)
        reversedText.append(word.reverse());

    System.out.println(reversedText);
}

/
=====
=====
=====
Output :

```

java MonthAndAverage.java

A dohtem sniatnoc eno ro erom stnemetats sdohteM era senituorbus taht etalupinam eht atad denifed yb eht ssalc dna, ni ynam sesac,edivorp ssecca ot taht atad

```

=====
=====
=====
Q24:AssignAref (pg. 169 A java book)
=====
=====
```

```

class AssignAref
{
    public static void main(String args[])
    {
        int i;
        int n1[] = new int[10];
        int n2[] = new int[10];

        for(i=0;i<10;i++)
            n2[i] = i;
        for(i=0;i<10;i++)
            n2[i] = -i;
        System.out.println("Here is n1 : ");
        for(i=0;i<10;i++)
            System.out.println(n1[i] + " ");
        System.out.println();

        System.out.println("Here is n2 : ");
        for(i=0;i<10;i++)
            System.out.println(n2[i] + " ");
        System.out.println();

        n2 = n1;

        System.out.println("Here is n2 after assignment : ");
    }
}
```

```
for(i=0;i<10;i++)
    System.out.println(n2[i] + " ");
System.out.println();

n2[3] = 99;
System.out.println("Here is n1 after assignment : ");
for(i=0;i<10;i++)
    System.out.println(n1[i] + " ");
System.out.println();
};

}
```

=====
=====
OUTPUT :

java assignAref.java

Here is n1 :

0
0
0
0
0
0
0
0
0
0

Here is n2 :

0
-1
-2
-3
-4
-5
-6
-7
-8
-9

Here is n2 after assignment :

0
0
0
0
0
0
0
0
0
0

0

Here is n1 after assignment :

```
0
0
0
99
0
0
0
0
0
0
0
```

=====

=====

Q25:Queue.java , QueueDemo.java (pg. 174 A java book)

=====

=====

```
class queue
{
    char q[];
    int putloc,getloc;

    queue(int size)
    {
        q = new char[size];
        putloc = getloc = 0;
    }
    void put(char ch)
    {
        if(putloc == q.length)
        {
            System.out.println(" -> Queue is full ");
            return;
        }
        q[putloc++] = ch;
        char get();
        {
            if(getloc == putloc)
            {
                System.out.println(" -> Queue is full ");
                return (char) 0 ;
            }
            return q[getloc++];
        }
    }
    class QueueDemo
    {
        public static void main(String[] args)
        {
```

```

queue bigQ = new queue(100);
queue smallQ = new queue(4);
char ch;
int i;

System.out.println("Using bigQ to store the alphabet");
for(i=0;i<26;i++)
    bigQ.put((char) ('A' + i));

System.out.println("Contents of bigQ : ");
for(i=0;i<26;i++)
{
    ch = bigQ.get();
    if( ch != (char) 0 )
        System.out.println(ch);
}
System.out.println("\n");

System.out.println("Using smallQ to store the alphabet");
for(i=0;i<5;i++)
{
    System.out.println(" Attempting to store " + (char) ('Z' - i));
    smallQ.put((char) ('Z' - i));
    System.out.println();
}
System.out.println("Contents of smallQ : ");
for(i=0;i<5;i++)
{
    ch = smallQ.get();
    if( ch != (char) 0 )
        System.out.println(ch);
}
//System.out.println("\n");
}

}

=====
=====

OUTPUT :

```

```

java queue.java
Using bigQ to store the alphabet
Contents of bigQ : ABCDEFGHIJKLMNOPQRSTUVWXYZ

```

Using smallQ to generate errors.

```

Attempting to store Z
Attempting to store Y
Attempting to store X
Attempting to store W
Attempting to store V - Queue is full

```

Contents of smallQ : ZYXW - Queue is empty

```
=====
=====
Q26:TryGeometry (pg. 263-Ivor Horton)
=====
=====

class TryGeometry
{
    public static void main(String[] args)
    {
        // Create two points and display them
        Point start = new Point(0.0, 1.0);
        Point end = new Point(5.0, 6.0);
        System.out.println("Points created are " + start + " and " + end);
        // Create two lines and display them
        Line line1 = new Line(start, end);
        Line line2 = new Line(0.0, 3.0, 3.0, 0.0);
        System.out.println("Lines created are " + line1 + " and " + line2);
        // Display the intersection
        System.out.println("Intersection is " + line2.intersects(line1));
        // Now move the end point of line1 and show the new intersection
        end.move(1.0, -5.0);
        System.out.println("Intersection is " + line1.intersects(line2));
    }
}
```

```
=====
=====
OUTPUT :
```

java trygeom.java

Points created are 0.0, 1.0 and 5.0, 6.0
 Lines created are (0.0, 1.0):(5.0, 6.0) and (0.0, 3.0):(3.0, 0.0)
 Intersection is 1.0, 2.0
 Intersection is 1.0, 2.0

```
=====
=====
Q27:Rabbit Out of hats (pg. 290-Ivor Horton)
=====
=====

MagicHat.java
```

```
import java.util.Random;
public class MagicHat {
    static int maxRabbits = 5;
```

```

static Random select = new Random();
public MagicHat(String hatName) {
    this.hatName = hatName;
    rabbits = new Rabbit[1+select.nextInt(maxRabbits)];

    for(int i = 0; i < rabbits.length; i++) {
        rabbits[i] = new Rabbit();
    }
}
// String representation of a hat
public String toString() {
    String hatString = "\n" + hatName + " contains:\n";
    for(Rabbit rabbit : rabbits) {
        hatString += " " + rabbit;
    }
    return hatString;
}
private String hatName;
private Rabbit rabbits[];

static class Rabbit {
    static private String[] rabbitNames = {"Flopsy", "Moppsy",
"Gnasher", "Thumper"};
    static private int[] rabbitNamesCount = new
int[rabbitNames.length];
    private String name;

    public Rabbit() {
        int index = select.nextInt(rabbitNames.length);
        name = rabbitNames[index] + (+
+rabbitNamesCount[index]); }
    public String toString() { return name; }
}
}

```

TryNestedClass.java

```

public class TryNestedClass {
    static public void main(String[] args) {
        // Create three magic hats and output them
        System.out.println(new MagicHat("Gray Topper"));
        System.out.println(new MagicHat("Black Topper"));
        System.out.println(new MagicHat("Baseball Cap"));
    }
}
=====
```

=====
=====
OUTPUT:

```
java TryNestedClass.java
```

Gray Topper contains:

Thumper1 Flopsy1 Thumper2 Thumper3

Black Topper contains:

Flopsey2 Thumper4

Baseball Cap contains:

Thumper5 Thumper6 Moppsy1

```
=====
=====
Q28:Nestedclassdemo.java (pg. 238 A java book)
=====
```

```
=====
class Outer{
    int nums[];
    Outer(int n[]){
        nums=n;
    }
    void analyze(){
        Inner inob = new Inner();
        System.out.println("Minimum : "+inob.min());
        System.out.println("Maximum: "+inob.max());
        System.out.println("Average : "+inob.avg());
    }
    class Inner{
        int min(){
            int m = nums[0];
            for(int i =1; i<nums.length; i++)
                if(nums[i] < m) m = nums[i];
            return m;
        }
        int max(){
            int m = nums[0];
            for(int i =1; i<nums.length; i++)
                if(nums[i] > m) m = nums[i];
            return m;
        }
        int avg(){
            int a = 0;
            for(int i =1; i<nums.length; i++)
                a += nums[i];
            return a/nums.length;
        }
    }
}
public class NestedClassDemo{
```

```

    public static void main(String args[]){
        int x[] = {3,2,1,5,6,9,7,8};
        Outer outob = new Outer(x);
        outob.analyze();
    }
}
=====
```

OUTPUT :

java NestedClassDemo.java

```

Minimum : 1
Maximum: 9
Average : 4
```

```

=====
=====
Q 29 : Implementing an interface
=====
=====
Conversions.java
```

```

public interface Conversions {
    double inchesToMillimeters (double inches);
    double ouncesToGrams(double ounces);
    double poundsToGrams(double pounds);
    double hpToWatts(double hp);
    double wattsToHP(double watts);
}
```

ConversionFactors.java

```

package conversions;
public class ConversionFactors {
    public static final double INCH_TO_MM = 25.4;
    public static final double OUNCE_TO_GRAM = 28.349523125;
    public static final double POUND_TO_GRAM = 453.5924;
    public static final double HP_TO_WATT = 745.7;
    public static final double WATT_TO_HP = 1.0/HP_TO_WATT;
}
```

TryConversions.java

```

import static conversions.ConversionFactors.*;
interface Conversions {
    double inchesToMillimeters (double inches);
    double ouncesToGrams(double ounces);
    double poundsToGrams(double pounds);
    double hpToWatts(double hp);
    double wattsToHP(double watts);
}
public class TryConversions implements Conversions {
    public double wattsToHP (double watts) {
        return watts*WATT_TO_HP;
    }
    public double hpToWatts (double hp) {
        return hp*HP_TO_WATT;
    }
    public double ouncesToGrams(double ounces) {
        return ounces*OUNCE_TO_GRAM;
    }
    public double poundsToGrams(double pounds) {
        return pounds*POUND_TO_GRAM;
    }
    public double inchesToMillimeters(double inches) {
        return inches*INCH_TO_MM;
    }
    public static void main(String args[]) {
        int myWeightInPounds = 180;
        int myHeightInInches = 75;
        TryConversions converter = new TryConversions();
        System.out.println("My weight in pounds: " +myWeightInPounds +
" \t-in grams: "+ (int)converter.poundsToGrams(myWeightInPounds));
        System.out.println("My height in inches: " + myHeightInInches +
" \t-in millimeters: " + (int)converter.inchesToMillimeters(myHeightInInches));
    }
}
=====
```

OUTPUT:

```
javac TryConversions.java
```

```
java TryConversions
My weight in pounds: 180      -in grams: 81646
My height in inches: 75      -in millimeters: 1905
=====
```

Q 30: Defining Interface

```
=====
=====
TV.java
```

```

import static java.lang.Math.max;
import static java.lang.Math.min;

public class TV implements RemoteControl {
    public TV(String make, int screensize) {
        this.make = make;
        this.screensize = screensize;
    }
    public boolean powerOnOff() {
        power = !power;
        System.out.println(make + " " + screensize + " inch TV power " +
(power ? "on.":"off."));
        return power;
    }
    public int volumeUp(int increment) {
        if(!power) {
            return 0;
        }
        volume += increment;
        volume = min(volume, MAX_VOLUME);
        System.out.println(make + " " + screensize + " inch TV volume level: " +
volume);
        return volume;
    }
    public int volumeDown(int decrement) {
        if(!power) {
            return 0;
        }
        volume -= decrement;
        volume = max(volume, MIN_VOLUME);
        System.out.println(make + " " + screensize + " inch TV volume level: " +
volume);
        return volume;
    }
    public void mute() {
        if(!power) {
            return;
        }
        volume = MIN_VOLUME;
        System.out.println(make + " " + screensize + " inch TV volume level: " +
volume);
    }
    public int setChannel(int newChannel) {
        if(!power) {
            return 0; // Nothing works
        }
        if(newChannel>=MIN_CHANNEL && newChannel<=MAX_CHANNEL)
            channel = newChannel;
            System.out.println(make + " " + screensize + " inch TV tuned to
channel: " + channel);
    }
}

```

```

        return channel;
    }
    public int channelUp() {
        if(!power) {
            return 0;
        }
        channel = channel<MAX_CHANNEL ? ++channel : MIN_CHANNEL;
        System.out.println(make + " " + screensize + " inch TV tuned to channel:
" + channel);
        return channel;
    }
    public int channelDown() {
        if(!power) {
            return 0;
        }
        channel = channel>MIN_CHANNEL ? --channel : MAX_CHANNEL;
        System.out.println(make + " " + screensize + " inch TV tuned to channel:
" + channel);
        return channel;
    }
    private String make = null;
    private int screensize = 0;
    private boolean power = false;
    private int MIN_VOLUME = 0;
    private int MAX_VOLUME = 100;
    private int volume = MIN_VOLUME;
    private int MIN_CHANNEL = 0;
    private int MAX_CHANNEL = 999;
    private int channel = 0;
}

```

VCR.java

```

import static java.lang.Math.max;
import static java.lang.Math.min;
public class VCR implements RemoteControl {
    public VCR(String make) {
        this.make = make;
    }
    public boolean powerOnOff() {
        power = !power;
        System.out.println(make + " VCR power "+ (power ? "on.":"off."));
        return power;
    }
    public int volumeUp(int increment) {
        if(!power) {
            return 0;
        }
        volume += increment;
        volume = min(volume, MAX_VOLUME);
        System.out.println(make + " VCR volume level: "+ volume);
    }
}

```

```

        return volume;
    }
    public int volumeDown(int decrement) {
        if(!power) {
            return 0;
        }
        volume -= decrement;
        volume = max(volume, MIN_VOLUME);
        System.out.println(make + " VCR volume level: "+ volume);
        return volume;
    }
    public void mute() {
        if(!power) {
            return;
        }
        volume = MIN_VOLUME;
        System.out.println(make + " VCR volume level: "+ volume);
    }
    public int setChannel(int newChannel) {
        if(!power) {
            return 0;
        }
        if(newChannel>=MIN_CHANNEL && newChannel<=MAX_CHANNEL) {
            channel = newChannel;
        }
        System.out.println(make + " VCR tuned to channel: "+ channel);
        return channel;
    }
    public int channelUp() {
        if(!power) {
            return 0;
        }
        channel = channel<MAX_CHANNEL ? ++channel : MIN_CHANNEL;
        System.out.println(make + " VCR tuned to channel: "+ channel);
        return channel;
    }
    public int channelDown() {
        if(!power) {
            return 0;
        }
        channel = channel>MIN_CHANNEL ? --channel : MAX_CHANNEL;
        System.out.println(make + " VCR tuned to channel: "+ channel);
        return channel;
    }
    private String make = null;
    private boolean power = false;
    private int MIN_VOLUME = 0;
    private int MAX_VOLUME = 100;
    private int volume = MIN_VOLUME;
    private int MIN_CHANNEL = 0;
    private int MAX_CHANNEL = 99;
    private int channel = 0;
}

```

```
}
```

remotecontrol.java

```
public interface RemoteControl {
    boolean powerOnOff();
    int volumeUp(int increment);
    int volumeDown(int decrement);
    void mute();
    int setChannel(int channel);      // Set the channel number and return it
    int channelUp();                // Returns new channel number int
    channelDown();                  // Returns new channel number
}
```

TryRemoteControl .java

```
import static java.lang.Math.random;
public class TryRemoteControl {
    public static void main(String args[]) {
        RemoteControl remote = null;
        for(int i = 0 ; i<5 ; i++) {
            if(random()<0.5) // Random choice of TV make and screen
size
                remote = new TV(random()<0.5 ? "Sony" : "Hitachi",
random()<0.5 ? 32 : 28);
            else
                remote = new VCR(random()<0.5 ? "Panasonic":
"JVC");
            remote.powerOnOff();
            remote.channelUp();
            remote.volumeUp(10);
        }
    }
}
```

```
=====
=====
```

OUTPUT :

```

java TryRemoteControl.java
Sony 32 inch TV power on.
Sony 32 inch TV tuned to channel: 1
Sony 32 inch TV volume level: 10
JVC VCR power on.
JVC VCR tuned to channel: 1
JVC VCR volume level: 10
Panasonic VCR power on.
Panasonic VCR tuned to channel: 1
Panasonic VCR volume level: 10
JVC VCR power on.
JVC VCR tuned to channel: 1
JVC VCR volume level: 10
Panasonic VCR power on.
Panasonic VCR tuned to channel: 1
Panasonic VCR volume level: 10

```

```
=====
=====
Q 31:Queue Interface.
=====
=====
interface ICharQ
{
    void put(char ch);
    char get();
}

class FixedQueue implements ICharQ
{
    private char q[];
    private int putloc,getloc;

    public FixedQueue(int size)
    {
        q = new char[size];
        putloc = getloc = 0;
    }
    public void put(char ch)
    {
        if(putloc == q.length)
        {
            System.out.println("-- Queue is full--");
            return;
        }
        q[putloc++] = ch;
    }

    public char get()
    {
```

```

        if(getloc == putloc)
    {
        System.out.println("-- Queue is empty--");
        return (char) 0;
    }
    return q[getloc++];
}

class CirculerQueue implements ICharQ
{
    private char q[];
    private int putloc, getloc;

    public CirculerQueue(int size)
    {
        q = new char[size + 1];
        putloc = getloc = 0;
    }
    public void put(char ch)
    {
        if(putloc+1 == getloc | ((putloc == q.length -1) & (getloc == 0)))
        {
            System.out.println("-- Queue is full--");
            return;
        }
        q[putloc++] = ch;
        if(putloc == q.length) putloc = 0;
    }

    public char get()
    {
        if(getloc == putloc)
        {
            System.out.println("-- Queue is empty--");
            return (char) 0;
        }

        char ch = q[getloc++];
        if(getloc == q.length) getloc = 0;
        return ch;
    }
}

class DynQueue implements ICharQ
{
    private char q[];
    private int putloc, getloc;

    public DynQueue(int size)

```

```

{
    q = new char[size + 1];
    putloc = getloc = 0;
}
public void put(char ch)
{
    if(putloc == q.length)
    {
        char t[] = new char[q.length * 2];

        for(int i=0;i<q.length; i++)
            t[i] = q[i];

        q=t;
    }
    q[putloc++] = ch;
}

public char get()
{
    if(getloc == putloc)
    {
        System.out.println("-- Queue is empty--");
        return (char) 0;
    }
    return q[getloc++];
}

}

public class IQDemo
{
    public static void main(String[] args)
    {
        FixedQueue q1 = new FixedQueue(10);
        DynQueue q2 = new DynQueue(5);
        CirculerQueue q3 = new CirculerQueue(10);
        ICharQ iQ;
        char ch;
        int i;

        iQ =q1;
        for(i=0;i<10;i++)
            iQ.put((char) ('A' + i));

        System.out.println("Content of fixed Queue");
        for(i=0;i<10;i++)
        {
            ch = iQ.get();
            System.out.print(ch);
        }
        System.out.println();
    }
}

```

```

iQ =q2;
for(i=0;i<10;i++)
    iQ.put((char) ('Z' - i));

System.out.println("Content of dynamic Queue");
for(i=0;i<10;i++)
{
    ch = iQ.get();
    System.out.print(ch);
}
System.out.println();

iQ =q3;
for(i=0;i<10;i++)
    iQ.put((char) ('A' + i));

System.out.println("Content of circular Queue");
for(i=0;i<10;i++)
{
    ch = iQ.get();
    System.out.print(ch);
}
System.out.println();

for(i=10;i<20;i++)
    iQ.put((char) ('A' + i));

System.out.println("Content of circular Queue");
for(i=0;i<10;i++)
{
    ch = iQ.get();
    System.out.print(ch);
}
System.out.println("\n Store and consume from " + " Circular Queue");

for(i=0;i<20;i++)
{
    iQ.put((char) ('A' + i));
    ch = iQ.get();
    System.out.print(ch);
}
}

=====
=====

OUTPUT :

```

Content of fixed Queue

ABCDEFGHIJ

Content of dynamic Queue

ZYXWVUTSRQ

Content of circular Queue

ABCDEFGHIJ

Content of circular Queue

KLMNOPQRST

Store and consume from Circular Queue

ABCDEFGHIJKLMNOPQRST

=====

=====

Q 32: Generic Functional Interface

=====

=====

```
interface SomeTest<T>{
    boolean test(T n,T m);
}
class GenericFunctionalInterfaceDemo{
    public static void main(String args[]){
        SomeTest<Integer> isFactor=(n,d) -> (n%d)==0;
        if(isFactor.test(10,2))
            System.out.println("2 is a factor of 10");
        System.out.println();

        SomeTest<Double> isFactorD = (n,d) -> (n%d)==0;
        if(isFactorD.test(212.0,4.0))
            System.out.println("4.0 is a factor of 212.0");
        System.out.println();

        SomeTest<String> isIn=(a,b) -> a.indexOf(b) !=-1;
        String str="Generic Functional Interface";

        System.out.println("Testing String: "+str);

        if(isIn.test(str,"face"))
            System.out.println("'face' is found.");
        else
            System.out.println("'face' not found.");
    }
}
```

=====

=====

OUTPUT :

java GenericFunctionalInterfaceDemo

2 is a factor of 10

4.0 is a factor of 212.0

Testing String: Generic Functional Interface
 'face' is found.

```
=====
=====
Q33: write an example of lambda_argument_demo
=====
=====

public class LambdaArgumentDemo {
    static String changeStr(StringFunc sf, String s) {
        return sf.func(s);
    }

    public static void main(String args[]) {
        String inStr = "Lambda Expression Expand Java";
        String outStr;

        System.out.println("Here is input string : " + inStr);

        StringFunc reverse = (str) -> {
            String result = "";
            for (int i = str.length()-1; i >= 0; i--)
                result += str.charAt(i);

            return result;
        };

        outStr = changeStr(reverse, inStr);
        System.out.println("The string reversed : " + outStr);

        outStr = changeStr((str) -> str.replace(' ', '-'), inStr);
        System.out.println("The string with spaces replaced : " + outStr);

        outStr = changeStr((str) -> {
            String result = "";
            char ch;

            for (int i = 0; i < str.length(); i++) {
                ch = str.charAt(i);
                if(Character.isUpperCase(ch))
                    result += Character.toLowerCase(ch);
                else
                    result += Character.toUpperCase(ch);
            }
            return result;
        }, inStr);
        System.out.println("The string in reversed case : " + outStr);
    }
}
```

```
}
```

```
interface StringFunc {
    String func(String str);
}
```

```
=====
=====
```

OUTPUT :

```
javac LambdaArgumentDemo.java
```

```
java LambdaArgumentDemo
```

```
Here is input string : Lambda Expression Expand Java
```

```
The string reversed : avaJ dnapxE noisserpxE adbmaL
```

```
The string with spaces replaced : Lambda-Expression-Expand-Java
```

```
The string in reversed case : IAMBDA eXPRESSION eXPAND jAVA
```

```
=====
=====
```

```
Q34: write an example lambdaexceptiondemo
```

```
=====
=====
```

```
import java.io.*;
interface MyIOAction{
    boolean ioAction(Reader rdr) throws IOException;
}
class LambdsExceptionDemo{
    public static void main(String args[])
    {
        double[] values = {1.0,2.0,3.0,4.0};
        MyIOAction myIO = (rdr) -> {
            int ch = rdr.read();
            return true;
        };
    }
}
```

```
=====
=====
```

OUTPUT :

```
javac LambdsExceptionDemo.java
```

```
java LambdsExceptionDemo
```

```
=====
=====
Q35: write a program to find square root of a number using package
=====
=====
```

ExtInterface.java

```
package extInterface;
public interface ExtInterface {
    public void method1();
    public void method2();

}
```

TestExtInterface.java

```
package extInterface;

import java.util.Scanner;

class TestExtInterface implements ExtInterface{
    public void method1() {
        System.out.println("Implementation of Method1");
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number to find the Square root in java:");
        double square = sc.nextDouble();
        double squareRoot = Math.sqrt(square);
        System.out.printf("Square root of number: %f is ->%f",square,squareRoot);

    }
    public void method2() {
        System.out.println("Implementation of method2");
    }
    public static void main(String[] args) {
        ExtInterface obj = new TestExtInterface();
        obj.method1();
    }
}
```

```
=====
=====
```

OUTPUT :

```
javac -d . ExtInterface.java

javac -d . TestExtInterface.java

java extInterface.TestExtInterface
```

Implementation of Method1

Enter number to find the Square root in java:

4

Square root of number: 4.000000 is -> 2.000000

=====

=====

Q36: write an example of arithmetic operations and method overriding using package

=====

=====

maths.java

package basicoperations;

```
public interface maths{
    public void add();
    public void sub();
    public void mul();
    public void div();
}
```

studen1.java

package basicoperations;
import java.util.Scanner;

```
public class studen1 implements maths
{
    @Override
    public void add()
    {
        Scanner kb = new Scanner(System.in);
        System.out.println("-----");
        System.out.println("Addition");
        System.out.println("-----");
        System.out.println("Enter any two integer values:");
        int no1 = kb.nextInt();
        int no2 = kb.nextInt();
        int sum = no1 + no2;
        System.out.println("sum of " + no1 + " and " + no2 + " is " + sum);
    }
    @Override
    public void sub()
    {
        Scanner kb = new Scanner(System.in);
        System.out.println("-----");
        System.out.println("Subtraction");
        System.out.println("-----");
        System.out.println("Enter any two integer values:");
        int no1 = kb.nextInt();
        int no2 = kb.nextInt();
    }
}
```

```

        int sum = no1 - no2;
        System.out.println("subtraction of " + no1 + " and " + no2 + " is " +
sum);
    }
    @Override
    public void mul()
    {
        Scanner kb = new Scanner(System.in);

        System.out.println("-----");
        System.out.println("Multiplication");
        System.out.println("-----");
        System.out.println("Enter any two integer values:");
        int no1 = kb.nextInt();
        int no2 = kb.nextInt();
        int sum = no1 * no2;
        System.out.println("multiplication of " + no1 + " and " + no2 + " is " +
sum);
    }
    @Override
    public void div()
    {
        Scanner kb = new Scanner(System.in);

        System.out.println("-----");
        System.out.println("Division");
        System.out.println("-----");
        System.out.println("Enter any two integer values:");
        int no1 = kb.nextInt();
        int no2 = kb.nextInt();
        int sum = no1 / no2;
        System.out.println("division of " + no1 + " and " + no2 + " is " + sum);
    }
    public static void main(String[] args)
    {
        studen1 student = new studen1();
        student.add();
        student.sub();
        student.mul();
        student.div();
    }
}

```

=====

=====

OUTPUT :

javac -d . maths.java

```
javac -d . studen1.java  
java basicoperations.studen1
```

Addition

Enter any two integer values:
5
7
sum of 5 and 7 is 12

Substraction

Enter any two integer values:
8
9
substration of 8 and 9 is -1

Multiplication

Enter any two integer values:
2
5
multiplication of 2 and 5 is 10

Division

Enter any two integer values:
4
7
division of 4 and 7 is 0

```
=====  
=====  
Q37: write an example of package  
=====  
=====
```

ClassA.java

```
package MCA;  
public class classA implements solution {  
    public void Hello()  
    {  
        System.out.println("Hello world");  
    }  
    public void Welcome()  
    {  
        System.out.println("Welcome MCAIII");  
    }  
  
    public static void main(String[] args)
```

```
{
    classA test = new classA();
    test.Hello();
    test.Welcome();
}
}
```

Solution.java

```
package MCA;
public interface solution{
    public void Hello();
    public void Welcome();
}
```

```
=====
=====
```

OUTPUT :

javac -d . solution.java

javac -d . ClassA.java

```
java MCA.ClassA
Hello world
Welcome MCAIII
```

```
=====
=====
```

Q38: write an example of book package

```
=====
=====
```

books.java

```
package bookpack;

public class books
{
    //these are protected members
    protected String title;
    protected String author;
    protected int pubDate;

    public books(String t, String a, int d)
    {
        title = t;
        author = a;
        pubDate = d;
    }
}
```

```

public void show()
{
    System.out.println(title);
    System.out.println(author);
    System.out.println(pubDate);
    System.out.println();
}
}

```

UseBooks.java

```

package bookpackext;
import bookpack.*;

class UseBook
{
    public static void main(String args[])
    {
        books book[] = new books[5];
        book[0] = new books("Java: A Beginner's Guide","Schildt",2014);

        book[1] = new books("Java: The Complete Reference","Schildt",2014);

        book[2] = new books("The Art of Java","Schildt and Holmes",2003);

        book[3] = new books("Red Storm Raising","Clancy",1986);

        book[4] = new books("On The Road","Kerouac",1955);

        for(int index = 0; index < book.length;index++) book[index].show();
    }
}

```

ProtectDemo.java

```

package bookpackext;

class ExtBook extends bookpack.books
{
    private String publisher;
    public ExtBook(String t,String a, int d,String p)
    {
        super(t,a,d);
        publisher = p;
    }

    public void show()
    {
        super.show();
        System.out.println(publisher);
        System.out.println();
    }
}

```

```

public String getPublisher() { return publisher; }
public void setPublisher(String p) { publisher = p; }

public String getTitle() { return title; }
public void setTitle(String t) { title = t; }

public String getAuthor() { return author; }
public void setAuthor(String a) { author = a; }

public int getPubDate() { return pubDate; }
public void setPubDate(int d) { pubDate = d; }

}

class ProtectDemo
{
    public static void main(String[] args)
    {
        ExtBook books[] = new ExtBook[5];
        books[0] = new ExtBook("Java: A Beginner's
Guide","Schildt",2014,"McGraw-Hill Professional");

        books[1] = new ExtBook("Java: The Complete
Reference","Schildt",2014,"McGraw-Hill Professional");

        books[2] = new ExtBook("The Art of Java","Schildt and
Holmes",2003,"McGraw-Hill Professional");

        books[3] = new ExtBook("Red Storm Raising","Clancy",1986,"Putham");

        books[4] = new ExtBook("On The Road","Kerouac",1955,"Viking");

        for(int index = 0; index < books.length; index++)
        {
            books[index].show();
        }

        System.out.println("Showing all books by schildt.");
        for(int index=0;index<books.length;index++)
        {
            if(books[index].getAuthor() == "Schildt")
            {
                System.out.println(books[index].getTitle());
            }
            //books[index].title = "test title"; error by accessing protected
        }
    }
}

=====
=====
```

OUTPUT :

```
javac -d . books.java
```

```
javac -d . UseBook.java
```

```
javac -d . ProtectDemo.java
```

```
java bookpackext.ProtectDemo
```

Java: A Beginner's Guide

Schildt

2014

McGraw-Hill Professional

Java: The Complete Reference

Schildt

2014

McGraw-Hill Professional

The Art of Java

Schildt and Holmes

2003

McGraw-Hill Professional

Red Storm Raising

Clancy

1986

Putham

On The Road

Kerouac

1955

Viking

Showing all books by schildt.

Java: A Beginner's Guide

Java: The Complete Reference

```
=====
```

```
=====
```

Q39: write an example of series program

```
=====
```

```
=====
```

Series.java

```
public interface Series
{
    int getNext();
    void reset();
    void setStart(int x);
}

SeriesDemo.java

class ByTwos implements Series
{
    int start;
    int val;

    ByTwos() {
        start = 0;
        val = 0;
    }

    public int getNext() {
        val += 2;
        return val;
    }

    public void reset()
    {
        val = start;
    }

    public void setStart(int x)
    {
        start = x;
        val = x;
    }
}

class ByThrees implements Series
{
    int start;
    int val;

    ByThrees() {
        start = 0;
        val = 0;
    }

    public int getNext() {
        val += 3;
        return val;
    }

    public void reset()
```

```

{
    val = start;
}

public void setStart(int x)
{
    start = x;
    val = x;
}
}

class SeriesDemo
{
    public static void main(String args[])
    {
        ByTwos twoOb = new ByTwos();
        ByThrees threeOb = new ByThrees();
        Series ob;

        for(int i=0; i < 5; i++)
        {
            ob = twoOb;
            System.out.println("Next ByTwos value is " + ob.getNext());

            ob = threeOb;
            System.out.println("Next ByThrees value is " + ob.getNext());
        }

        /*for(int i=0; i < 5; i++)
            System.out.println("Next value is " + ob.getNext());

        System.out.println("\nResetting");
        ob.reset();
        for(int i=0;i<5;i++)
            System.out.println("Next value is "+ob.getNext());*/

        System.out.println("\nStarting at 100");
        ob.setStart(100);
        for(int i=0;i<5;i++)
            System.out.println("Next value is "+ob.getNext());
    }
}
=====
```

OUTPUT :

javac Series.java

javac SeriesDemo.java

```
java SeriesDemo
Next ByTwos value is 2
Next ByThrees value is 3
Next ByTwos value is 4
Next ByThrees value is 6
Next ByTwos value is 6
Next ByThrees value is 9
Next ByTwos value is 8
Next ByThrees value is 12
Next ByTwos value is 10
Next ByThrees value is 15
```

=====

=====

Q40: write an example of lambda displayable

=====

```
=====
public class LambdaDisplayable implements displayble {
    public void display(String msg){
        System.out.println(msg);
    }
    public static void main(String[] args) {
        LambdaDisplayable dis = new LambdaDisplayable();
        dis.display("Welcome to the World.");
    }
}

interface displayble {
    void display(String msg);
}
```

=====

=====

OUTPUT :

```
javac LambdaDisplayable.java
```

```
java LambdaDisplayable
Welcome to the World.
```

=====

=====

Q41: write an example of lambda using functional interface

=====

=====

=====

```
interface FuncInterface
{
```

```

void abstractFun(int x);

default void normalFun()
{
    System.out.println("Hello");
}
}

class Test
{
    public static void main(String[] args)
    {
        FuncInterface fobj = (int x) -> System.out.println(2*x);
        fobj.abstractFun(5);
    }
}

```

=====

=====

OUTPUT :

javac Test.java

java Test
10

=====

=====

Q42: write an example of calculator using lambda

=====

=====

```

public class Calculator
{
    interface IntegerMath
    {
        int operation(int a,int b);
    }

    public int OperateBinary(int a,int b,IntegerMath op){
        return op.operation(a,b);
    }

    public static void main(String[] args)
    {
        Calculator myApp = new Calculator();
        IntegerMath addition = (a,b) -> a + b;
        IntegerMath subtraction = (a,b) -> a-b;
        System.out.println("40 + 2 = " + myApp.OperateBinary(40,2,addition));
    }
}

```

```

        System.out.println("20 - 10 = " +
myApp.OperateBinary(20,10,substration));
    }
}

```

=====

=====

OUTPUT :

```
javac Calculator.java
```

```
java Calculator
40 + 2 = 42
20 - 10 = 10
```

=====

=====

Q43: write a program to implement uber using interface, anonymous class and lambda expression

=====

=====

Uber1.java

```
@FunctionalInterface
interface Cab {
    void bookCab();
}

class UberX1 implements Cab {
    public void bookCab() {
        System.out.println("UberX Booked!! Arriving Soon !!");
    }
}
```

```
public class Uber1 {
    public static void main(String[] args) {
        Cab cab = new UberX1();
        cab.bookCab();
    }
}
```

Uber2.java

```
@FunctionalInterface
interface Cab {
    void bookCab();
}

public class Uber2 {
```

```

public static void main(String[] args) {
    Cab cab = new Cab() {

        @Override
        public void bookCab() {
            System.out.println("UberX Booked!! Arriving Soon !!");
        }
    };

    cab.bookCab();
}
}

```

Uber3.java

```

@FunctionalInterface
interface Cab {
    void bookCab();
}

public class Uber3 {
    public static void main(String[] args) {
        Cab cab = () -> System.out.println("UberX Booked!! Arriving Soon !!");
        cab.bookCab();
    }
}

```

OUTPUT :

javac Uber1.java

java Uber1
UberX Booked!! Arriving Soon !!

javac Uber2.java

java Uber2
UberX Booked!! Arriving Soon !!

javac Uber3.java

java Uber3
UberX Booked!! Arriving Soon !!

Assignment - 1

=====
=====
NAME : Ajinkya Rathod
ROLL NO : 30
CLASS : MCA 3
SUBJECT : JAVA
=====
=====

ASSIGNMENT - 1

=====
=====
Q1 : Write a program to generate first 50 prime numbers .(Use all types of loops)
=====
=====

FOR LOOP :

```
import java.lang.*;
import java.util.Scanner;

class p1{
    public static void main(String args[]){

        Scanner scn = new Scanner(System.in);

        int n;
        int isPrime = 1;
        int num = 3;

        for ( int count = 1 ; count <=50 ; ) {
            for ( int j = 2 ; j <= Math.sqrt(num) ; j++ ) {
                if ( num%j == 0 ) {
                    isPrime = 0;
                    break;
                }
            }
            if ( isPrime != 0 )
            {
                System.out.print(num + " ");
                count++;
            }
            isPrime = 1;
            num++;
        }
    }
}
```

```
}
```

```
=====
=====
OUTPUT :
```

```
java p1.java
3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107
109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223
227 229 233
=====
```

```
=====
WHILE LOOP :
```

```
import java.lang.*;
import java.util.Scanner;

class p1{
    public static void main(String args[]){

        Scanner scn = new Scanner(System.in);

        int n;
        int isPrime = 1;
        int num = 3;
        int count = 0;

        while ( count !=50 ) {
            for ( int j = 2 ; j <= Math.sqrt(num) ; j++ ) {
                if ( num%j == 0 ) {
                    isPrime = 0;
                    break;
                }
            }
            if ( isPrime != 0 )
            {
                System.out.print(num + " ");
                count++;
            }
            isPrime = 1;
            num++;
        }
    }
}
```

```
=====
=====
OUTPUT :
```

```
java p1_b
```

```
3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107
109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223
227 229 233
=====
```

using Do While Loop:

```
import java.lang.*;
import java.util.Scanner;

class p1{
    public static void main(String args[]){

        Scanner scn = new Scanner(System.in);

        int n;
        int isPrime = 1;
        int num = 3;
        int count = 0;

        do{
            for ( int j = 2 ; j <= Math.sqrt(num) ; j++ ) {
                if ( num%j == 0 ) {
                    isPrime = 0;
                    break;
                }
            }
            if ( isPrime != 0 )
            {
                System.out.print(num + " ");
                count++;
            }
            isPrime = 1;
            num++;
        } while(count != 50) ;
    }
}
```

```
=====
```

=====

OUTPUT:

java p1_c

```
3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107
109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223
227 229 233
=====
```

```
=====
```

Q2 : 2. Write a program that accepts two command line arguments of type double, multiply them and display their product.

```
=====
=====
public class p2{
    public static void main(String args[]){
        double first_no = Double.parseDouble(args[0]);
        double second_no = Double.parseDouble(args[1]);

        double mul = first_no * second_no;

        System.out.println("Product = " + mul);
    }
}
```

```
=====
=====
```

OUTPUT:

```
java p2.java 4 5
Product = 20.0
```

```
=====
=====
=====*
```

Q3 : Write a program that accepts radius of a circle and display area of circle.

```
=====
=====
public class p3{
    public static void main(String args[])
    {
        int r = Integer.parseInt(args[0]);

        double area = 3.14 * r * r;

        System.out.println("Area of Circle = " + area);
    }
}
```

OUTPUT :

```
java p3.java 2
Area of Circle = 12.56
```

```
=====
=====
=====
```

OUTPUT:

Q5 : Write a program to accept one command line argument , convert it to double and then display the

square root.

```
=====
=====
import java.lang.Math;
public class p5{
    public static void main(String args[])
    {
        double no = Double.parseDouble(args[0]);
        double sqroot = Math.sqrt(no);
        System.out.println("Sqrt is "+ sqroot);
    }
}
```

=====

=====

OUTPUT:

```
java p5.java 25
Sqrt is 5.0
```

=====

=====

Q6: Write a program that prints numbers between 17 and 100 that are evenly divisible by 17.

```
=====
=====
public class p6{
    public static void main(String args[])
    {
        for(int i = 17; i <= 100; i++)
        {
            if(i % 17 == 0)
            {
                System.out.println(i + " ");
            }
        }
    }
}
```

=====

=====

OUTPUT:

```
java p6.java
17
34
51
68
85
```

=====

=====

Q7 : Write a program that displays all the possible factors of a particular number.

```
=====
=====
import java.util.Scanner;
```

```

public class p7{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Positive Number For Factors:");

        int no = sc.nextInt();

        System.out.print("Factors = ");
        for(int i = 1; i <= no; i++)
        {
            if(no % i == 0)
            {
                System.out.print(i + " ");
            }
        }
    }
=====
=====

OUTPUT:
java p7.java
Enter Positive Number For Factors:
50
Factors = 1 2 5 10 25 50
=====
=====

Q8 : Write a program that prints numbers from 1 to 100 using five columns.
=====
=====

public class p8{
    public static void main(String args[]){
        //int n = 1;
        for(int i = 1; i <= 20; i++)
        {
            System.out.println(i + " " + (i+20) + " " + (i+40) + " " + (i+60) +
" " + (i+80));
        }
    }
=====
=====

OUTPUT:
java p8.java
1 21 41 61 81
2 22 42 62 82
3 23 43 63 83
4 24 44 64 84
5 25 45 65 85
6 26 46 66 86
7 27 47 67 87
8 28 48 68 88
9 29 49 69 89

```

10	30	50	70	90
11	31	51	71	91
12	32	52	72	92
13	33	53	73	93
14	34	54	74	94
15	35	55	75	95
16	36	56	76	96
17	37	57	77	97
18	38	58	78	98
19	39	59	79	99
20	40	60	80	100

Q9 : Write a program that accepts one command line argument,convert it to integer and decrement its value in a loop.Sound the bell when value reaches 0.(Bell character:\v0007).

```
import java.util.Scanner;
public class p9{
    public static void main(String args[]){
        int no = Integer.parseInt(args[0]);
        for(int i = no ; i >= 0; i--)
        {
            System.out.println(no);
            no--;
            if(no == 0)
            {
                System.out.print("\007");//for beep sound
            }
        }
    }
}
```

OUTPUT:

```
java p9.java 5  
5  
4  
3  
2  
1  
0
```

Q10 : Write a program that declares a class Person. It should have instance variable to record name , age and salary of type string , integer and float . Set and display these properties.

```
import java.util.Scanner;
```

```

class person{
    int age;
    float salary;
    String name;

    Scanner sc = new Scanner(System.in);

    void getdata()
    {
        System.out.println("Enter Name :");
        name = sc.nextLine();
        System.out.println("Enter Age :");
        age = sc.nextInt();
        System.out.println("Enter Salary :");
        salary = sc.nextFloat();
    }

    void display()
    {
        System.out.println("\n\t\t====* DISPLAY DATA ===*\n");
        System.out.println("Name = " +name);
        System.out.println("Age = " +age);
        System.out.println("Salary = " +salary);
    }
}

public class p10{
    public static void main(String args[]){
        person p = new person();
        p.getdata();
        p.display();
    }
}
=====
=====

OUTPUT :
java p10
Enter Name :
thindo
Enter Age :
21
Enter Salary :
300000

====* DISPLAY DATA ===*

Name = thindo
Age = 21
Salary = 300000.0

```

Assignment - 2

Q 1 : Write a java program to add two matrices of same size.

```
import java.util.Scanner;
class a2_p1{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int r,c,i,j,r1,c1;

        System.out.println("Enter rows and columns for 1st Matrix:");
        System.out.print("Enter Rows :");
        r = sc.nextInt();
        System.out.print("Enter Cols :");
        c = sc.nextInt();

        System.out.println("Enter rows and columns for 2nd Matrix:");
        System.out.print("Enter Rows :");
        r1 = sc.nextInt();
        System.out.print("Enter Cols :");
        c1 = sc.nextInt();

        if(r == r1 && c == c1)
        {
            int first[][] = new int[r][c];//declare matrix
            int second[][] = new int[r1][c1];
            int add[][] = new int[r1][c1];

            System.out.println("Enter Elements for Matrix 1 :");
            for(i = 0; i < r; i++)
            {
                for(j = 0; j < c; j++)
                {
                    first[i][j] = sc.nextInt();
                }
            }

            System.out.println("Enter Elements for Matrix 2 :");
            for(i = 0; i < r1; i++)
            {
                for(j = 0; j < c1; j++)
```

```

    {
        second[i][j] = sc.nextInt();
    }
}

System.out.println(" \t\t: Matrix 1 :");
for(i = 0; i < r; i++)
{
    for(j = 0; j < c; j++)
    {
        System.out.print(first[i][j] + "   ");
    }
    System.out.println();
}

System.out.println(" \t\t: Matrix 2 :");
for(i = 0; i < r; i++)
{
    for(j = 0; j < c; j++)
    {
        System.out.print(second[i][j] + "   ");
    }
    System.out.println();
}
for(i = 0; i < r; i++)
{
    for(j = 0; j < c1; j++)
    {
        for(int k = 0; k < c; k++)
        {
            add[i][j] = first[i][j] + second[i][j];
        }
    }
}
System.out.println("\t\t== Matrix After Addition ==\n");
for(i = 0; i < r; i++)
{
    for(j = 0; j < c; j++)
    {
        System.out.print(add[i][j] + "   ");
    }
    System.out.println();
}
}
else
{
    System.out.println("Enter same rows and cols for both matrix..!");
}
=====
=====

OUTPUT :

```

```

java a2_p1
Enter rows and colmns for 1st Matrix:
Enter Rows :2
Enter Cols :3
Enter rows and colmns for 2nd Matrix:
Enter Rows :2
Enter Cols :3
Enter Elements for Matrix 1 :
1
2
3
4
5
6
Enter Elements for Matrix 2 :
1
2
3
4
5
6
      : Matrix 1 :
1 2 3
4 5 6
      : Matrix 2 :
1 2 3
4 5 6
===== Matrix After Addition =====

2 4 6
8 10 12
=====
=====
Q 2 : Write a java program to get environment variables USERNAME,TEMPNAME and
PATH.
=====
=====
class a2_p2{
    public static void main(String args[]){
        System.out.println("\nEnvironment Variable PATH : ");
        System.out.println(System.getenv("PATH"));
        System.out.println("\nEnvironment Variable TEMPNAME : ");
        System.out.println(System.getenv("TEMPNAME"));
        System.out.println("\nEnvironment Variable USERNAME : ");
        System.out.println(System.getenv("USERNAME"));
    }
}
=====
=====
OUTPUT:
java a2_p2

```

Environment Variable PATH :

```
/home/ajinzrathod/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/lib/jvm/java-15-oracle/bin:/usr/lib/jvm/java-15-oracle/db/bin:/home/ajinzrathod/.local/bin/
```

Environment Variable TEMPNAME :

```
null
```

Environment Variable USERNAME :

```
ajinzrathod
```

```
=====
=====
```

Q 3 : Write an application that defines a Circle class with two constructors .The first form accepts a double value that represents the radius of the circle .This constructor assumes that the circle is centered at origin .The second form accepts three double values .The first two arguments defines the coordinates of the center and the third argument defines the radius .Create 10 objects of the Circle type and save them in the array. Randomly select a radius between 1 and 10 cm for each circle created .After all circles have been created display each circle and total area of all circles.

```
=====
=====
```

```
class circle
{
    double area;
    public static int totalarea=0;

    circle(double radius)
    {
        area=3.14*radius*radius;
        System.out.println("Area = " + Math.round(area));
        totalarea+=area;
    }

    circle(double x,double y,double radius)
    {

        area=(3.14*radius*radius);
        System.out.println("Area for Origin(" + Math.round(x) + "," + Math.round(y) + ") = " + Math.round(area));
        totalarea+=area;
    }
}

class a2_p3
{
    public static void main(String[] args)
```

```

{
    circle[] cir=new circle[10];

    for(int i=0; i<5; i++)
        cir[i]=new circle(((Math.random()*10)+1));

    System.out.println();

    for(int i=5; i<10; i++)
        cir[i]=new circle(((Math.random()*10)+1),((Math.random()*10)+1),
((Math.random()*10)+1));

    System.out.println();

    System.out.println("Total Area is: " + cir[0].totalarea);
}
}
=====
```

OUTPUT:

```

java a2_p3
Area = 167
Area = 134
Area = 325
Area = 74
Area = 25

Area for Origin(8,5) = 38
Area for Origin(4,4) = 86
Area for Origin(9,4) = 176
Area for Origin(6,4) = 31
Area for Origin(6,5) = 76
```

Total Area is: 1127

```

=====
=====
```

Q 4 : Write a program to remove duplicate characters from the string.

```

=====
=====
```

```

class a2_p4
{
    public static void main(String[] args)
    {
        String name=System.console().readLine("Enter a String: ");
        char[] arr=name.toCharArray();
        char[] newstr=new char[50];
        int ctr=0;

        for(int i=0; i<arr.length; i++)
        {
```

```

        int flag=0;
        for(int j=0; j<i; j++)
        {
            if(arr[i]==arr[j])
            {
                flag=1;
            }
        }

        if(flag==0)
        {
            newstr[ctr]=arr[i];
            ctr++;
        }
    }

    String str=new String(newstr);
    System.out.println("\nFinal String: " + str);
}
=====
=====

OUTPUT:

```

java a2_p4
Enter a String: ajinky

Final String: ajinky

=====
=====
Q 5 : Write a program to print the pattern using for loop.

1 1 1
2 4 8
3 9 27
.....
10 100 1000

=====
=====
=====
class a2_p3{
 public static void main(String args[]){
 int i;
 for(i = 1; i <= 10; i++)
 {
 System.out.println(i + " " + (i*i) + " " + (i*i*i));
 }
 }
}
=====
=====

OUTPUT:

java a2_p3

```

1 1 1
2 4 8
3 9 27
4 16 64
5 25 125
6 36 216
7 49 343
8 64 512
9 81 729
10 100 1000
=====
=====
```

Q 6 : Write a java program to accept distance in metres , time in hours ,minutes and seconds .Calculate
the speed in m/sec ,km/hr and miles/hr .(1 mile=1609 m).

```
=====
=====
```

```
import java.util.Scanner ;
```

```

public class a2_p6 {
    public static void main(String[] args) {
        double dis_m ;
        double dis_km;
        double dis_mile;
        int hr;
        int min;
        int sec;
        int t_m;
        int t_s;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Distance in meter : ");
        dis_m = sc.nextDouble();

        dis_km = dis_m / 1000 ;
        dis_mile = dis_m / 1609;

        System.out.println("Distance in meter = " + dis_m);
        System.out.println("Distance in k_m = " + dis_km);
        System.out.println("Distance in mile = " + dis_mile);

        System.out.println("Enter Hours : ");
        hr = sc.nextInt();
        System.out.println("Enter Minutes : ");
        min = sc.nextInt();
        System.out.println("Enter Seconds : ");
        sec = sc.nextInt();

        t_m = (hr*60) + min ;
        t_s = (t_m * 60)+ sec ;
```

```

        System.out.println("The Speed in m/sec is : " + dis_m / t_s);
        System.out.println("The Speed in km/hr is : " + dis_km / hr);
        System.out.println("The Speed in mile/hr is : " + dis_mile /hr);

    }

=====
=====


```

OUTPUT:

```

java a2_p6
Enter Distance in meter :
1000
Distance in meter = 1000.0
Distance in k_m = 1.0
Distance in mile = 0.6215040397762586
Enter Hours :
1
Enter Minutes :
0
Enter Seconds :
0
The Speed in m/sec is : 0.2777777777777778
The Speed in km/hr is : 1.0
The Speed in mile/hr is : 0.6215040397762586

=====


```

Q 7 :Write a program to get java version, runtime version ,java vendor name , java vendor url.

```

=====
=====

class a2_p7
{
    public static void main(String[] args)
    {
        System.out.println("Java Version: " + System.getProperty("java.version"));
        System.out.println("Vendor Name: " + System.getProperty("java.vendor"));
        System.out.println("Vendor URL: " + System.getProperty("java.vendor.url"));
    }
}

=====


```

OUTPUT:

```

java a2_p7
Java Version: 14.0.2


```

Vendor Name: Oracle Corporation
 Vendor URL: <https://java.oracle.com/>

```
=====
=====
Q 8 :Create an abstract class "Monster" which is extended by three classes "Ware wolf
","Zombie" and
"Vampire".Create 6 types of Monsters in single dimension array and print them.
=====

import java.util.*;

abstract class monster
{
    abstract void getdata();
}

class warewolf extends monster
{
    void getdata()
    {
        System.out.println("I'm WareWolf");
    }
}

class zombie extends monster
{
    void getdata()
    {
        System.out.println("I'm Zombie");
    }
}

class vampire extends monster
{
    void getdata()
    {
        System.out.println("I'm Vampire");
    }
}

public class a2_p8
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        monster[] m1=new monster[6];
        String mtype;

        for(int i=0; i<6; i++)
        {
            System.out.println("What type of monster you are?
(warewolf/zombie/vampire): ");
        }
    }
}
```

```

mtype=sc.nextLine();

if(mtype.equals("warewolf"))
{
    m1[i]=new warewolf();
}
else if(mtype.equals("zombie"))
{
    m1[i]=new zombie();
}
else if(mtype.equals("vampire"))
{
    m1[i]=new vampire();
}
else
{
    System.out.println("\nYou're not a monster!");
}

System.out.println("\n\nDisplaying Data:-\n");
for(int i=0; i<6; i++)
{
    m1[i].getdata();
}

}
=====
=====

OUTPUT:
```

```

java a2_p8
What type of monster you are? (warewolf/zombie/vampire):
zombie
What type of monster you are? (warewolf/zombie/vampire):
warewolf
What type of monster you are? (warewolf/zombie/vampire):
zombie
What type of monster you are? (warewolf/zombie/vampire):
vampire
What type of monster you are? (warewolf/zombie/vampire):
zombie
What type of monster you are? (warewolf/zombie/vampire):
vampire
```

Displaying Data:-

```

I'm Zombie
I'm WareWolf
I'm Zombie
I'm Vampire
```

I'm Zombie
I'm Vampire

Q 9:Write a program to accept a string from user and a point from where you want to print next three words. Print the old and the new string.

Old String = "The quick brown fox jumps over the lazy dog"

New String = "brown fox jumps"

```
class a2_p9
{
    public static void main(String[] args)
    {
        int flag=0,count=0;
        String org=System.console().readLine("Enter a String: ");
        String[] arr=org.split(" ");

        String find=System.console().readLine("Enter a string to print next 3 words:
");

        for(int i=0; i<arr.length; i++)
        {
            if(arr[i].equals(find))
            {
                flag=1;
            }

            if(flag==1 && count<3)
            {
                System.out.println("\nString " + (count+1) + ": " + arr[i]);
                count++;
            }
        }
    }
}
```

OUTPUT:

java a2_p9

Enter a String: lorem ipsum dolor set amet

Enter a string to print next 3 words: ipsum

String 1: dolor

String 2: set

String 3: amet

```
=====
=====
Q 10:Convert Minutes into years and days
For Eg. Input => 3456789
Output => 6 Years and 210 days
=====
=====
import java.util.*;

class a2_p10
{
    public static void main(String[] args)
    {
        int years=0,days=0,min;

        Scanner sc=new Scanner(System.in);

        System.out.print("Enter minutes: ");
        min=sc.nextInt();

        days=min/1440; // 1 Day= 1440 mins

        while(days>=365)
        {
            years++;
            days=days-365;
        }

        System.out.println("\n" + years + " years and " + days + " days");
    }
}
```

=====

=====

OUTPUT:

```
java a2_p10
Enter minutes: 3456789
```

6 years and 210 days

=====

=====

Q 11(a):Create a class Inventory having item_id, Description, price as member variable.
Create
necessary constructors and methods.

Create a class Bill_Item having quantity, amount(amount=quantity*price) as member variable.

Create necessary constructors and methods.

Create a Main class which will create 5 instances. Generate invoice.

=====

=====

```
import java.util.Scanner;

class inventory {
    int item_id ;
    String name ;
    double price ;
    inventory(){
    }
}

class bill_item extends inventory{
    int quantity ;
    double amount ;

    public void get_data(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter item id :");
        item_id = sc.nextInt();
        System.out.println("Enter name of the product :");
        name = sc.next();
        System.out.println("Enter the price of the product :");
        price = sc.nextDouble();
        System.out.println("Enter the quantity of the product :");
        quantity = sc.nextInt();
    }

    public void calculate_amount(){
        amount = (quantity * price );
    }

    public void generate_bill(){
        System.out.print(item_id + "\t\t");
        System.out.print(name+ "\t");
        System.out.print(price + "\t");
        System.out.print(quantity + "\t\t");
        System.out.print(amount+ "\t");
    }
}

public class a2_p11_a{
    public static void main(String args[]){
        int i = 0;
        bill_item[] b1 = new bill_item[10];
        for(i = 0 ; i < 2 ;i++){
            b1[i] = new bill_item();
            b1[i].get_data();
            b1[i].calculate_amount();
        }
    }
}
```

```

        System.out.println("Item id " + "\t" + "Name" + "\t" + "Price" + "\t" +
"Quantity" + "\t" + "Amount" );
        for(i = 0 ; i < 2 ;i++){
            b1[i].generate_bill();
            System.out.println();
        }
    }
}
=====
```

OUTPUT:

```

java a2_p11_a
Enter item id :
1
ENter name of the product :
TV
Enter the price of the product :
250000
Enter the quantity of the product :
2
Enter item id :
2
ENter name of the product :
AC
Enter the price of the product :
30000
Enter the quantity of the product :
3
Item id      Name   Price  Quantity     Amount
1           TV     250000.0    2       500000.0
2           AC     30000.0 3       90000.0
=====
```

Q 11(b):Take a string from user. Find the occurrences of a word. And print that word in reverse order.

```

=====
=====
```

```

class a2_p11_b
{
    public static void main(String[] args)
    {
        int count=0;
        String org=System.console().readLine("\nEnter a String: ");
        String[] arr=org.split(" ");

        String find=System.console().readLine("\nEnter a string to find total
occurrences: ");
        String rev = new StringBuffer(find).reverse().toString();

        for(int i=0; i<arr.length; i++)
        {
```

```

        if(arr[i].equals(find))
        {
            count++;
        }
    }

    System.out.println("\nThe word " + find + " occurs total " + count + " times.");
    System.out.println("\nReversed String: " + rev);
}

=====
=====
```

OUTPUT:

```
java a2_p11_b
```

Enter a String: Quick Brown ABC Fox ABC Jumps Over ABC

Enter a string to find total occurrences: ABC

The word ABC occurs total 3 times.

Reversed String: CBA

```
=====
=====
```

Q 12(a):Write a program Addition of two binary numbers

Example:

Input: Binary number 1: 101

Binary number 2: 011

Output: 1000

```
=====
=====
```

import java.util.*;

```

public class a2_p12_a
{
    public static void main(String[] args)
    {
        long b1, b2;
        int i = 0, carry = 0;

        int[] sum = new int[10];

        Scanner scanner = new Scanner(System.in);

        System.out.print("\nEnter first binary number: ");
        b1 = scanner.nextLong();
```

```
System.out.print("\nEnter second binary number: ");
b2 = scanner.nextLong();

while (b1 != 0 || b2 != 0)
{
    sum[i++] = (int)((b1 % 10 + b2 % 10 + carry) % 2);
    carry = (int)((b1 % 10 + b2 % 10 + carry) / 2);
    b1 = b1 / 10;
    b2 = b2 / 10;
}
if (carry != 0)
{
    sum[i++] = carry;
}
--i;

System.out.print("\nOutput: ");

while (i >= 0)
{
    System.out.print(sum[i--]);
}

System.out.print("\n");
}
```

java a2 p12 a

Enter first binary number: 11

Enter second binary number: 11

Output: 110

Q 13(b):(B) Print following pattern

Q 12(b).

Java
Java

J a a v v a a
11 a a a a a a v v a a a a

jj dddd v
jj a a v a a

=====
=====
=====
=====
OUTPUT:

```
=====
=====
Q13(a):Print Face
+"""""+
[| o o |]
| ^ |
| '-' |
+----+
=====
=====
public class a2_p13_a {
    public static void main(String[] args)
    {
        System.out.println(" +\"\\\"\\\"\\\"");
        System.out.println("[| o o |]");
        System.out.println(" | ^ | ");
        System.out.println(" | '-' | ");
        System.out.println(" +----+ ");
    }
}
```

```
=====
=====
OUTPUT:
    java a2_p13_a
+"""""+
[| o o |]
| ^ |
| '-' |
+----+
```

```
=====
=====
Q13(b):B.Print following pattern.
```

```
1
01
101
0101
10101
010101
=====
=====
class a2_p13_b{
    public static void main(String args[])
    {
        int i,j;
        for(i = 0; i <= 5; i++)
        {
            for(j = 0 ; j <= i; j++)
            {
```

```

        if((i+j) % 2 == 0 ){
            System.out.print(1);
        }
        else{
            System.out.print(0);
        }
    }
    System.out.println();
}
}

=====
=====

OUTPUT:
```

```
java a2_p13_b
1
01
101
0101
10101
010101
```

Q14:Create an employee class which has attributes : empid , fname , lname, salary, designation, bonus

.It has all the necessary constructors (at least 2) and methods.It must have one abstract method-
calculate_bonus().

- Create 2 subclasses :Clerk & Manager
- Clerk has the additional attribute-No. of hrs worked ,
- Manager has the attribute – No. of projects managed
- Both the subclasses must have necessary constructor (at least 2) and methods
- Create an array of five employees ,read the details and display the details
- Calculate the bonus on the following criteria :
- Clerk -No. of hrs worked>250 then for each extra hour 200 Rs Bonus
- Manager -No. of projects managed *1000

```
=====
=====

abstract class employee {
    int emp_id ;
    String fname;
    String lname ;
    float salary;
    String designation;
    float bonus;

    employee(int emp_id,String fname,String lname,float salary,String designation){
        this.emp_id = emp_id ;
```

```

this.fname = fname;
this.lname = lname;
this.salary = salary;
this.designation = designation ;
}

abstract void calculate_bonus();
abstract void display();
}

class clerk extends employee{

    int no_of_hrs_worked ;

    clerk(int emp_id,String fname,String lname,float salary,String designation, int c_n_w)
    {
        super(emp_id,fname,lname,salary,designation);
        this.no_of_hrs_worked = c_n_w ;
    }
    void calculate_bonus(){
        if(no_of_hrs_worked > 250 ){
            this.bonus = ( no_of_hrs_worked - 250 ) * 200 ;
        }
        else{
            this.bonus = 0 ;
        }
    }
    void display(){
        calculate_bonus();
        System.out.println("id is : " + this.emp_id);
        System.out.println("first name is : " + this.fname);
        System.out.println("last name is : " + this.lname);
        System.out.println("designation is : " + this.designation);
        System.out.println("basic salary is : " + this.salary);
        System.out.println("bonus is : " + this.bonus);
        System.out.println("total salary is : " + (this.salary + this.bonus));
    }
}

class manager extends employee{
    int no_project;
    manager(int emp_id,String fname,String lname,float salary,String designation, int n_o_p){
        super(emp_id,fname,lname,salary,designation);
        this.no_project = n_o_p ;
    }

    void calculate_bonus(){
        this.bonus = (no_project * 1000) ;
    }

    void display(){
        calculate_bonus();
    }
}

```

```

        System.out.println("id is : " + this.emp_id);
        System.out.println("first name is : " + this.fname);
        System.out.println("last name is : " + this.lname);
        System.out.println("designation is : " + this.designation);
        System.out.println("basic salary is : " + this.salary);
        System.out.println("bonus is : " + this.bonus);
        System.out.println("total salary is : " + (this.salary+this.bonus)));
    }
}

public class a2_p14{
    public static void main(String[] args){

        employee[] emp=new employee[5];

        emp[0] = new clerk(1,"Thindo","Ninja",580000,"Clerk",100);
        emp[1] = new clerk(2,"Thindi","Rathore",568000,"Manager",2);
        emp[2] = new clerk(3,"Ajinkya","Rathod",159000,"Clerk",300);
        emp[3] = new manager(1,"Ajinki","Jinki",752000,"Manager",3);
        emp[4] = new manager(2,"Aju","Bhai",650000,"Manager",3);

        for(int i=0; i<5; i++)
        {
            System.out.println("\n");
            emp[i].display();
        }
    }
=====
=====

OUTPUT:

java a2_p14

id is : 1
first name is : Thindo
last name is : Ninja
designation is : Clerk
basic salary is : 580000.0
bonus is : 0.0
total salary is : 580000.0

id is : 2
first name is : Thindi
last name is : Rathore
designation is : Manager
basic salary is : 568000.0
bonus is : 0.0
total salary is : 568000.0

id is : 3
first name is : Ajinkya
last name is : Rathod

```

```
designation is : Clerk  
basic salary is : 159000.0  
bonus is : 10000.0  
total salary is : 169000.0
```

```
id is : 1  
first name is : Ajinki  
last name is : Jinki  
designation is : Manager  
basic salary is : 752000.0  
bonus is : 3000.0  
total salary is : 755000.0
```

```
id is : 2  
first name is : Aju  
last name is : Bhai  
designation is : Manager  
basic salary is : 650000.0  
bonus is : 3000.0  
total salary is : 653000.0
```

Hands on: Classwork - 4

Q. ExcDemo1

```
public class ArrayException {  
    public static void main(String[] args) {  
        int nums[] = new int[5];
```

```

try {
    System.out.println("Before exception...");
    nums[7] = 70;
    System.out.println("This is not displayed...");
} catch (ArrayIndexOutOfBoundsException ae) {
    System.out.println(ae.getMessage());
}
System.out.println("After Catch...");
}
}

```

OUTPUT

```

$ javac ArrayException.java && java ArrayException
Before exception...
Index 7 out of bounds for length 5
After Catch...

```

Q. ExcDemo2.java

```

class ExcTest {
    static void genException() {
        int nums[] = new int[4];
        nums[7] = 10;
        System.out.println("not displayed");
    }
}

public class ExcDemo2 {
    public static void main(String[] args) {
        try {
            ExcTest.genException();
        } catch (ArrayIndexOutOfBoundsException e) {
            e.printStackTrace();
        }
        System.out.println("After Catch...");
    }
}

```

OUTPUT

```

$ javac ExcDemo2.java && java ExcDemo2
java.lang.ArrayIndexOutOfBoundsException: Index 7 out of bounds for length 4
    at ExcTest.genException(ExcDemo2.java:4)
    at ExcDemo2.main(ExcDemo2.java:12)
After Catch...

```

Q. Throw demo or rethrow demo (341-343)(perform any one , one-oracle press)

```

class ReThrow {
    public static void genException() {
        int numer[] = { 4, 8, 16, 32, 64, 128, 256, 512 };
        int denom[] = { 2, 0, 4, 4, 0, 8 };

        for (int i = 0; i < numer.length; i++) {
            try {
                System.out.println(numer[i] + " / " + denom[i] + " is " + numer[i] /
denom[i]);
            } catch (ArithmaticException e) {
                System.out.println("Cannot divide by Zero");
            } catch (ArrayIndexOutOfBoundsException e) {
                System.out.println("No matching element found");
                throw e;
            }
        }
    }
}

public class ReThrowDemo {
    public static void main(String[] args) {
        try {
            ReThrow.genException();
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Fatal Error - Program terminated");
        }
    }
}

```

OUTPUT

```

$ javac ReThrowDemo.java && java ReThrowDemo
4 / 2 is 2
Cannot divide by Zero
16 / 4 is 4
32 / 4 is 8
Cannot divide by Zero
128 / 8 is 16
No matching element found
Fatal Error - Program terminated

```

Q. FinallyDemo (oracle press 347)

```

class UseFinally {
    public static void genException(int what) {
        int t;
        int nums[] = new int[2];

```

```

System.out.println("Receiveing " + what);
try {
    switch (what) {
        case 0:
            t = 10 / what;
            break;
        case 1:
            nums[4] = 4;
            break;
        case 2:
            return;
    }
} catch (ArithmaticException e) {
    System.out.println(e.getMessage());
} catch (ArrayIndexOutOfBoundsException e) {
    System.out.println(e.getMessage());
} finally {
    System.out.println("Leaving Try");
}
}

public class FinallyDemo {
    public static void main(String[] args) {
        for (int i = 0; i < 3; i++) {
            UseFinally.genException(i);
            System.out.println();
        }
    }
}

```

OUTPUT

```

$ javac FinallyDemo.java && java FinallyDemo
Receiveing 0
/ by zero
Leaving Try

```

```

Receiveing 1
Index 4 out of bounds for length 2
Leaving Try

```

```

Receiveing 2
Leaving Try

```

Q. ThrowsDemo (oracle press 349)

```

import java.io.IOException;

public class ThrowsDemo {

```

```

public static char prompt(String str) throws IOException {
    System.out.print(str + " : ");
    return (char) System.in.read();
}

public static void main(String[] args) {
    char ch;

    try {
        ch = prompt("Enter a letter");
    } catch (IOException e) {
        System.out.println("I/O Exception occurred");
        ch = 'X';
    }
    System.out.println("You pressed " + ch);
}
}

```

OUTPUT

```

$ javac ThrowsDemo.java && java ThrowsDemo
Enter a letter : 9
You pressed 9

```

Q. MultiCatch feature (oracle press 350)

```

public class MultiCatch {
    public static void main(String[] args) {
        int a = 88, b = 0;
        int result;
        char ch[] = { 'A', 'B', 'C' };

        for (int i = 0; i < 2; i++) {
            try {
                if (i == 0)
                    result = a / b;
                else
                    ch[5] = 'X';
            } catch (ArithmaticException | ArrayIndexOutOfBoundsException e) {
                System.out.println(e.getMessage());
            }
        }
        System.out.println("After multi Catch");
    }
}

```

OUTPUT

```

$ javac MultiCatch.java && java MultiCatch
/ by zero

```

Index 5 out of bounds for length 3
After multi Catch

Q. Custom ExceptionDemo (oracle press 353)

```
class NonIntResultException extends Exception {
    int n, d;

    NonIntResultException(int i, int j) {
        n = i;
        d = j;
    }

    public String toString() {
        return "Result of " + n + " / " + d + " is non integer";
    }
}

public class CustomExceptionDemo {
    public static void main(String[] args) {
        int numer[] = { 4, 8, 15, 32, 64, 128, 256, 512 };
        int denom[] = { 2, 0, 4, 4, 0, 8 };

        for (int i = 0; i < numer.length; i++) {
            try {
                if (numer[i] % 2 != 0)
                    throw new NonIntResultException(numer[i], denom[i]);

                System.out.println(numer[i] + " / " + denom[i] + " is " + numer[i] / denom[i]);
            } catch (ArithmaticException e) {
                System.out.println("Cannot divide by Zero");
            } catch (ArrayIndexOutOfBoundsException e) {
                System.out.println("No matching element found");
            } catch (NonIntResultException e) {
                System.out.println(e);
            }
        }
    }
}
```

OUTPUT

```
$ javac CustomExceptionDemo.java && java CustomExceptionDemo
4 / 2 is 2
Cannot divide by Zero
Result of 15 / 4 is non integer
32 / 4 is 8
Cannot divide by Zero
128 / 8 is 16
```

No matching element found
 No matching element found

Q. Add exception to queue (oracle press 354)

```
class QueueFullException extends Exception {
    int size;

    QueueFullException(int s) {
        size = s;
    }

    public String toString() {
        return "\nQueue is full. Maximum size is " + size;
    }
}

class QueueEmptyException extends Exception {
    QueueEmptyException() {
    }

    public String toString() {
        return "\nQueue is Empty.";
    }
}

class Queue {
    private char[] q;
    private int putloc, getloc;

    Queue(int size) {
        q = new char[size];
        putloc = getloc = 0;
    }

    void put(char ch) throws QueueFullException {
        if (putloc == q.length) {
            throw new QueueFullException(q.length);
        }
        q[putloc++] = ch;
    }

    char get() throws QueueEmptyException {
        if (getloc == putloc)
            throw new QueueEmptyException();

        return q[getloc++];
    }
}
```

```

public class QueueDemo {
    public static void main(String[] args) throws Exception {
        Queue bigQ = new Queue(100);
        Queue smallQ = new Queue(4);
        char ch;
        int i;

        for (i = 0; i < 26; i++)
            bigQ.put((char) ('A' + i));

        for (i = 0; i < 26; i++) {
            ch = bigQ.get();
            if (ch != (char) 0)
                System.out.print(ch);
        }

        System.out.println("\n");

        System.out.println("Using smallQ to generate errors.");
        for (i = 0; i < 5; i++) {
            System.out.println("Attempting to store " + (char) ('Z' - i));

            smallQ.put((char) ('Z' - i));

            System.out.println();
        }
        System.out.println();

        System.out.println("Contents of SmallQ: ");
        for (i = 0; i < 5; i++) {
            ch = smallQ.get();
            if (ch != (char) 0)
                System.out.print(ch);
        }
    }
}

```

OUTPUT

```
$ javac QueueDemo.java && java QueueDemo
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```

Using smallQ to generate errors.
Attempting to store Z
Attempting to store Y
Attempting to store X
Attempting to store W
Attempting to store V
Exception in thread "main"
Queue is full. Maximum size is 4
    at Queue.put(QueueDemo.java:33)
    at QueueDemo.main(QueueDemo.java:68)

```

Q. Try it out (ivor horton 366)

```

class ZeroDivideException extends Exception {
    private int index = -1;

    public ZeroDivideException() {}

    public ZeroDivideException(String s) {
        super(s);
    }

    public ZeroDivideException(int index) {
        super("/ by zero");
        this.index = index;
    }

    public int getIndex() {
        return index;
    }
}

public class TryItOut {
    public static int divide(int[] array, int index) throws ZeroDivideException {
        try {
            System.out.println("First try block in divide() entered");
            array[index + 2] = array[index] / array[index + 1];
            System.out.println("Code at end of first try block in divide()");
            return array[index + 2];
        } catch (ArithmaticException e) {
            System.out.println("Arithmatic exception caught in divide()");
            throw new ZeroDivideException(index + 1);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Index-out-of-bounds index exception caught in
divide()");
        }
        System.out.println("Executing code after try block in divide()");
        return array[index + 2];
    }

    public static void main(String[] args) {
        int[] x = { 10, 5, 0 };
        try {
            System.out.println("First try block in main()entered");
            System.out.println("result = " + divide(x, 0));
            x[1] = 0;
            System.out.println("result = " + divide(x, 0));
            x[1] = 1;
        }
    }
}

```

```

        System.out.println("result = " + divide(x, 1));
    } catch (ZeroDivideException e) {
        int index = e.getIndex();
        if (index > 0) {
            x[index] = 1;
            x[index + 1] = x[index - 1];
            System.out.println("Zero divisor corrected to " + x[index]);
        }
    } catch (ArithmaticException e) {
        System.out.println("Arithmatic exception caught in main()");
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("Index-out-of-bounds exception caught in main()");
    }
    System.out.println("Outside first try block in main()");
}
}

```

OUTPUT

```

$ javac TryItOut.java && java TryItOut
First try block in main()entered
First try block in divide() entered
Code at end of first try block in divide()
result = 2
First try block in divide() entered
Arithmatic exception caught in divide()
Zero divisor corrected to 1
Outside first try block in main()

```

Q. Java 368(show file) program

```

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

public class ShowFile {
    public static void main(String[] args) {

        int i;
        FileInputStream fin;

        if (args.length != 1) {
            System.out.println("Usgae: ShowFile File");
            return;
        }

        try {
            fin = new FileInputStream(args[0]);
        } catch (FileNotFoundException e) {
            System.out.println(e.getMessage());
        }
    }
}

```

```
        return;
    }

try {
    do {
        i = fin.read();
        if (i != -1)
            System.out.print((char) i);
    } while (i != -1);
} catch (IOException e) {
    System.out.println(e.getMessage());
}

try {
    fin.close();
} catch (IOException e) {
    System.out.println(e.getMessage());
}
}
```

OUTPUT

```
$ javac ShowFile.java && java ShowFile file1.txt  
Some data in file
```

Q. Help -> 10.2(pg 394)

```
import java.io.*;  
  
class Help {  
    String helpfile;  
  
    Help(String fname) {  
        helpfile = fname;  
    }  
  
    boolean helpOn(String what) {  
        int ch;  
        String topic, info;  
  
        try (BufferedReader helprdr = new BufferedReader(new  
FileReader(helpfile))) {  
            do {  
                ch = helprdr.read();  
                if (ch == '#') {  
                    topic = helprdr.readLine();  
                    if (what.compareTo(topic) == 0) {  
                        return true;  
                    }  
                }  
            } while (ch != -1);  
        } catch (IOException e) {}  
        return false;  
    }  
}
```

```

        do {
            info = helprdr.readLine();
            if (info != null)
                System.out.println(info);
        } while ((info != null) && (info.compareTo("") != 0));
        return true;
    }
}
} while (ch != 1);
} catch (IOException e) {
    System.out.println(e.getMessage());
    return false;
}
return false;
}

String getSelection() {
    String topic = "";
    BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
    System.out.print("Enter topic: ");
    try {
        topic = br.readLine();
    } catch (IOException e) {
        System.out.println(e.getMessage());
    }
    return topic;
}
}

public class FileHelp {
    public static void main(String[] args) {
        Help hlpobj = new Help("helpfile.txt");
        String topic;

        System.out.println("try the help system." + "Enter 'stop' to end");

        do {
            topic = hlpobj.getSelection();

            if (!hlpobj.helpOn(topic))
                System.out.println("Topic not found");

        } while (topic.compareTo("stop") != 0);
    }
}

```

OUTPUT:

```
$ javac FileHelp.java && java FileHelp
try the help system.Enter 'stop' to end
Enter topic: topic2
this is help on topic 2
Enter topic: topic1
this is help on topic 1
```

Q. Write A String

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.FileNotFoundException;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;

public class WriteAString {
    public static void main(String[] args) {
        String phrase = new String("Garbage in, garbage out\n");
        String dirname = "C:/Beg Java Stuff";
        String filename = "charData.txt";
        File dir = new File(dirname);

        if (!dir.exists()) {
            if (!dir.mkdir()) {
                System.out.println("Cannot create directory: " + dirname);
                System.exit(1);
            }
        } else if (!dir.isDirectory()) {
            System.err.println(dirname + " is not a directory");
            System.exit(1);
        }
        File aFile = new File(dir, filename);
        FileOutputStream outputFile = null;
        try {
            outputFile = new FileOutputStream(aFile, true);
            System.out.println("File stream created successfully.");
        } catch (FileNotFoundException e) {
            e.printStackTrace(System.err);
        }

        FileChannel outChannel = outputFile.getChannel();
        ByteBuffer buf = ByteBuffer.allocate(1024);
        System.out.println("New buffer: position = " + buf.position() + "\tLimit = "
+ buf.limit() + "\tcapacity = "
+ buf.capacity());

        for (char ch : phrase.toCharArray()) {
```

```

        buf.putChar(ch);
    }
    System.out.println("Buffer after loading: position = " + buf.position() + "\n"
tLimit = " + buf.limit()
        + "\tcapacity = " + buf.capacity());
    buf.flip();
    System.out.println("Buffer after flip: position = " + buf.position() + "\n"
tLimit = " + buf.limit()
        + "\tcapacity = " + buf.capacity());

try {
    outChannel.write(buf);
    outputFile.close();
    System.out.println("Buffer contents written to file.");
} catch (IOException e) {
    e.printStackTrace(System.err);
}
System.exit(0);
}
}

```

OUTPUT

```

$ javac WriteAString.java && java WriteAString
File stream created successfully.
New buffer: position = 0      Limit = 1024      capacity = 1024
Buffer after loading: position = 48      Limit = 1024      capacity = 1024
Buffer after flip: position = 0 Limit = 48      capacity = 1024
Buffer contents written to file.

```

Q. Write A String As Bytes

```

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.FileNotFoundException;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;

public class WriteAStringAsBytes {
    public static void main(String[] args) {
        String phrase = new String("Garbage in, garbage out\n");
        String dirname = "C:/Beg Java Stuff";
        String filename = "byteData.txt";
        File aFile = new File(dirname, filename);

        FileOutputStream file = null;
        try {
            file = new FileOutputStream(aFile, true);
        } catch (FileNotFoundException e) {

```

```
        e.printStackTrace(System.err);
    }
    FileChannel outChannel = file.getChannel();
    ByteBuffer buf = ByteBuffer.allocate(phrase.length());
    byte[] bytes = phrase.getBytes();
    buf.put(bytes);
    buf.flip();
    try {
        outChannel.write(buf);
        file.close();
    } catch (IOException e) {
        e.printStackTrace(System.err);
    }
}
```

Q. Proverbs

```
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.FileNotFoundException;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;

public class WriteProverbs {
    public static void main(String[] args) {
        String dirName = "c:/Beg Java Stuff";
        String fileName = "Proverbs.txt";
        String[] sayings = { "Indecision maximizes flexibility.", "Only the mediocre
are always at their best.",
                    "A little knowledge is a dangerous thing.", "Many a mickle makes a
muckle.", "Who begins too much achieves little.", "Who knows most says
least.", "A wise man sits on the hole in his carpet." };
        File aFile = new File(dirName, fileName);
        FileOutputStream outputFile = null;
        try {
            outputFile = new FileOutputStream(aFile, true);
        } catch (FileNotFoundException e) {
            e.printStackTrace(System.err);
            System.exit(1);
        }
        FileChannel outChannel = outputFile.getChannel();
        int maxLength = 0;
        for (String saying : sayings) {
            if (maxLength < saying.length())
                maxLength = saying.length();
        }
        ByteBuffer buffer = ByteBuffer.allocate(maxLength);
        for (String saying : sayings)
            buffer.put(saying.getBytes());
        buffer.flip();
        outChannel.write(buffer);
        outChannel.close();
    }
}
```

```

        maxLength = saying.length();
    }
    ByteBuffer buf = ByteBuffer.allocate(2 * maxLength + 4);

    try {
        for (String saying : sayings) {
            buf.putInt(saying.length()).asCharBuffer().put(saying);
            buf.position(buf.position() + 2 * saying.length()).flip();
            outChannel.write(buf);
            buf.clear();
        }
        outputFile.close();
        System.out.println("Proverbs written to file.");
    } catch (IOException e) {
        e.printStackTrace(System.err);
        System.exit(1);
    }
    System.exit(0);
}
}

```

OUTPUT

```
$ javac WriteProverbs.java && java WriteProverbs
Proverbs written to file.
```

Q. Use A Formatter

```

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.FileNotFoundException;
import java.nio.ByteBuffer;
import java.nio.CharBuffer;
import java.nio.channels.FileChannel;
import java.util.Formatter;

public class UsingAFormatter {
    public static void main(String[] args) {
        String[] phrases = { "Rome wasn't burned in a day.", "It's a bold mouse
that sits in the cat's ear.",
        "An ounce of practice is worth a pound of instruction." };
        String dirname = "C:/Beg Java Stuff";
        String filename = "Phrases.txt";
        File dir = new File(dirname);

        if (!dir.exists()) {
            if (!dir.mkdir()) {
                System.out.println("Cannot create directory: " + dirname);
                System.exit(1);
            }
        }
    }
}

```

```

        }
    } else if (!dir.isDirectory()) {
        System.err.println(dirname + " is not a directory");
        System.exit(1);
    }

    File aFile = new File(dir, filename);
    FileOutputStream outputFile = null;
    try {
        outputFile = new FileOutputStream(aFile, true);
        System.out.println("File stream created successfully.");
    } catch (FileNotFoundException e) {
        e.printStackTrace(System.err);
    }

    FileChannel outChannel = outputFile.getChannel();

    ByteBuffer buf = ByteBuffer.allocate(1024);
    System.out.println("\nByte buffer:");
    System.out.printf("position = %2d Limit = %4d capacity = %4d%n",
buf.position(), buf.limit(), buf.capacity());

    CharBuffer charBuf = buf.asCharBuffer();
    System.out.println("Char view buffer:");
    System.out.printf("position = %2d Limit = %4d capacity = %4d%n",
charBuf.position(), charBuf.limit(),
    charBuf.capacity());
    Formatter formatter = new Formatter(charBuf);

    int number = 0;
    for (String phrase : phrases) {
        formatter.format("%nProverb%3d: %s", ++number, phrase);
        System.out.println("\nView buffer after loading:");
        System.out.printf("position = %2d Limit = %4d capacity = %4d%n",
charBuf.position(), charBuf.limit(),
    charBuf.capacity());
        charBuf.flip();
        System.out.println("View buffer after flip:");
        System.out.printf("position = %2d Limit = %4d length = %4d%n",
charBuf.position(), charBuf.limit(),
    charBuf.length());
        buf.limit(2 * charBuf.length());
        System.out.println("Byte buffer after limit update:");
        System.out.printf("position = %2d Limit = %4d length = %4d%n",
buf.position(), buf.limit(),
    buf.remaining());

        try {
            outChannel.write(buf);
            System.out.println("Buffer contents written to file.");
            buf.clear();
            charBuf.clear();
        }
    }
}

```

```
        } catch (IOException e) {
            e.printStackTrace(System.err);
            System.exit(1);
        }
    }
try {
    outputFile.close();
}
catch (IOException e) {
    e.printStackTrace(System.err);
}
}
```

OUTPUT

```
$ javac UsingAFormatter.java && java UsingAFormatter  
File stream created successfully.
```

Byte buffer:

position = 0 Limit = 1024 capacity = 1024

Char view buffer:

position = 0 Limit = 512 capacity = 512

View buffer after loading:

position = 44 Limit = 512 capacity = 512

View buffer after flip:

position = 0 Limit = 44 length = 44

Byte buffer after limit update:

position = 0 Limit = 88 length = 88

Buffer contents written to file.

position = 63 Limit = 51

View buffer after flip:

position = 0 Limit =

Byte buffer after limit update:

Byte buffer after limit update:
position = 0 Limit = 126 length = 126

Buffer contents written to file.

View buffer after loading:

position = 67 Limit = 512 capacity = 512

View buffer after flip:

position = 0 limit = 67 length = 67

Byte buffer after limit update:

Byte buffer after limit update:
position = 0 limit = 134 length = 134

Buffer contents written to file

Q. Try It Out Copying File (creating backup files)

```

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.channels.FileChannel;

public class FileCopy {
    public static void main(String[] args) {
        if (args.length == 0) {
            System.out.println("No file to copy. Application usage is:\n" + "java -"
                    + "classpath . FileCopy \"filepath\"");
            System.exit(1);
        }
        File fromFile = new File(args[0]);
        if (!fromFile.exists()) {
            System.out.printf("File to copy, %s, does not exist.", "
                    fromFile.getAbsolutePath());
            System.exit(1);
        }
        File toFile = createBackupFile(fromFile);
        FileInputStream inFile = null;
        FileOutputStream outFile = null;
        try {
            inFile = new FileInputStream(fromFile);
            outFile = new FileOutputStream(toFile);
        } catch (FileNotFoundException e) {
            e.printStackTrace(System.err);
            assert false;
        }
        FileChannel inChannel = inFile.getChannel();
        FileChannel outChannel = outFile.getChannel();
        try {
            int bytesWritten = 0;
            long byteCount = inChannel.size();
            while (bytesWritten < byteCount) {
                bytesWritten += inChannel.transferTo(bytesWritten, byteCount -
                        bytesWritten, outChannel);
            }
            System.out.printf("File copy complete. %d bytes copied to %s%n",
                    byteCount, toFile.getAbsolutePath());
            inFile.close();
            outFile.close();
        } catch (IOException e) {
            e.printStackTrace(System.err);
            System.exit(1);
        }
        System.exit(0);
    }

    public static File createBackupFile(File aFile) {

```

```
aFile = aFile.getAbsoluteFile();
File parentDir = new File(aFile.getParent());
String name = aFile.getName();
int period = name.indexOf('.');
if (period == -1) {
    period = name.length();
}
String nameAdd = "_backup";
File backup = aFile;
while (backup.exists()) {
    name = backup.getName();
    backup = new File(parentDir, name.substring(0, period) + nameAdd +
name.substring(period));
    period += nameAdd.length();
}
return backup;
}
```

OUTPUT

```
$ javac FileCopy.java && java FileCopy "Beg_Java_Stuff/charData.txt"  
File copy complete. 48 bytes copied to D:\Study\MCA\SEM III\Java\Practice\  
Beg_Java_Stuff\charData_backup.txt
```

Q. Read And Write

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;

public class RandomReadWrite {
    public static void main(String[] args) {
        File aFile = new File("primes.bin");
        FileInputStream inFile = null;
        FileOutputStream outFile = null;
        try {
            inFile = new FileInputStream(aFile);
            outFile = new FileOutputStream(aFile, true);
        } catch (FileNotFoundException e) {
            e.printStackTrace(System.err);
            System.exit(1);
        }
        FileChannel inChannel = inFile.getChannel();
        FileChannel outChannel = outFile.getChannel();
        final int PRIMESREQUIRED = 10;
        ByteBuffer buf = ByteBuffer.allocate(8);
```

```
long[] primes = new long[PRIMESREQUIRED];
int index = 0;
final long REPLACEMENT = 99999L;
try {
    final int PRIMECOUNT = (int) inChannel.size() / 8;
    System.out.println("Prime count = " + PRIMECOUNT);
    for (int i = 0; i < PRIMESREQUIRED; i++) {
        index = 8 * (int) (PRIMECOUNT * Math.random());
        inChannel.read(buf, index);
        buf.flip();
        primes[i] = buf.getLong();
        buf.flip();
        buf.putLong(REPLACEMENT);
        buf.flip();
        outChannel.write(buf, index);
        buf.clear();
    }
    int count = 0;
    for (long prime : primes) {
        System.out.printf("%12d", prime);
        if (++count % 5 == 0) {
            System.out.println();
        }
    }
    inFile.close();
    outFile.close();
} catch (IOException e) {
    e.printStackTrace(System.err);
    System.exit(1);
}
System.exit(0);
}
```

OUTPUT

359	107	383	109	7
173	443	337	17	113

Q. UseThreads

```
class MyThread implements Runnable {  
    String thrdName;  
  
    MyThread(String name) {  
        thrdName = name;  
    }  
  
    public void run() {  
        System.out.println(thrdName + " starting.");  
    }  
}
```

```

try {
    for (int count = 0; count < 10; count++) {
        Thread.sleep(5000);
        System.out.println("In " + thrdName + ", count is " + count);
    }
} catch (InterruptedException exc) {
    System.out.println(thrdName + " interrupted.");
}

System.out.println(thrdName + " terminating.");
}

}

class UseThreads {
    public static void main(String[] args) {
        System.out.println("Main thread starting.");

        MyThread mt = new MyThread("Child #1");

        Thread newThrd = new Thread(mt);

        newThrd.start();

        for (int i = 0; i < 50; i++) {
            System.out.println(".\n" + newThrd.isAlive());
            try {
                Thread.sleep(500);
            } catch (InterruptedException exc) {
                System.out.println("Main thread interrupted.");
            }
            newThrd.interrupt();
        }
        System.out.println("\nMain thread ending.");
    }
}

```

Q. Sum Array Synchronization

```

class SumArray {
    private int sum;

    synchronized int sumArray(int[] nums) {
        sum = 0;

        for (int i = 0; i < nums.length; i++) {
            sum += nums[i];
            System.out.println("Running total for " +
Thread.currentThread().getName() + " is " + sum);

        try {

```

```

        Thread.sleep(10);
    } catch (InterruptedException exc) {
        System.out.println("Thread interrupted!");
    }
}
return sum;
}

class MyThread implements Runnable {
    Thread thrd;
    static SumArray sa = new SumArray();
    int[] a;
    int answer;

    MyThread(String name, int[] nums) {
        thrd = new Thread(this, name);
        a = nums;
        thrd.start();
    }

    public void run() {
        int sum;

        System.out.println(thrd.getName() + " starting...");

        answer = sa.sumArray(a);
        System.out.println("Sum for " + thrd.getName() + " is " + answer);

        System.out.println(thrd.getName() + " terminating...");
    }
}

public class SumArraySync {
    public static void main(String[] args) {
        int[] a = { 1, 2, 3, 4, 5 };

        MyThread mt1 = new MyThread("Child #1", a);
        MyThread mt2 = new MyThread("Child #2", a);

        try {
            mt1.thrd.join();
            mt2.thrd.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted!");
        }
    }
}

```

OUTPUT

```
$ javac SumArraySync.java && java SumArraySync
```

```

Child #2 starting...
Child #1 starting...
Running total for Child #2 is 1
Running total for Child #2 is 3
Running total for Child #2 is 6
Running total for Child #2 is 10
Running total for Child #2 is 15
Running total for Child #1 is 1
Sum for Child #2 is 15
Child #2 terminating...
Running total for Child #1 is 3
Running total for Child #1 is 6
Running total for Child #1 is 10
Running total for Child #1 is 15
Sum for Child #1 is 15
Child #1 terminating...

```

Q. Class TickTock

```

class TickTock {
    String state;

    synchronized void tick(boolean running) {
        if (!running) {
            state = "ticked";
            notify();
            return;
        }

        System.out.println("Tick ");

        state = "ticked";
        notify();

        try {
            while (!state.equals("tocked"))
                wait();
        } catch (InterruptedException e) {
            System.out.println("Thread interrupted");
        }
    }

    synchronized void tock(boolean running) {
        if (!running) {
            state = "tocked";
            notify();
            return;
        }

        System.out.println("Tock ");
    }
}

```

```

state = "tocked";
notify();

try {
    while (!state.equals("ticked"))
        wait();
} catch (InterruptedException e) {
    System.out.println("Thread interrupted");
}
}

class MyThread implements Runnable {
    Thread thrd;
    TickTock ttOb;

    MyThread(String name, TickTock tt) {
        thrd = new Thread(this, name);
        ttOb = tt;
        thrd.start();
    }

    public void run() {
        if (thrd.getName().compareTo("Tick") == 0) {
            for (int i = 0; i < 5; i++)
                ttOb.tick(true);
            ttOb.tick(false);
        } else {
            for (int i = 0; i < 5; i++)
                ttOb.tock(true);
            ttOb.tock(false);
        }
    }
}

public class ThreadCom {
    public static void main(String[] args) {
        TickTock t1 = new TickTock();
        MyThread mt1 = new MyThread("Tick", t1);
        MyThread mt2 = new MyThread("Tock", t1);

        try {
            mt1.thrd.join();
            mt2.thrd.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted...");
        }
    }
}

```

OUTPUT

```
$ javac ThreadCom.java && java ThreadCom
Tick
Tock
Tick
Tock
Tick
Tock
Tick
Tock
Tick
Tock
```

Q. Suspend.java

```
class MyThread implements Runnable {
    Thread thrd;

    boolean suspended;
    boolean stopped;

    MyThread(String name) {
        thrd = new Thread(this, name);
        suspended = false;
        stopped = false;
        thrd.start();
    }

    // this is the entry point for thread
    public void run() {
        System.out.println(thrd.getName() + " Starting.");
        try {
            for (int i = 1; i < 1000; i++) {
                System.out.println(i + " ");
                if ((i % 10) == 0) {
                    System.out.println();
                    Thread.sleep(250);
                }
                synchronized (this) {
                    while (suspended) {
                        wait();
                    }
                    if (stopped)
                        break;
                }
            }
        } catch (InterruptedException e) {
            System.out.println(thrd.getName() + " Interrupted.");
        }
        System.out.println(thrd.getName() + " exiting.");
    }
}
```

```

}

synchronized void mystop() {
    stopped = true;
    suspended = false;
    notify();
}

synchronized void mysuspend() {
    suspended = true;
}

synchronized void myresume() {
    suspended = false;
    notify();
}
}

class Suspend {
    public static void main(String[] args) {
        MyThread ob1 = new MyThread("My Thread");
        try {
            Thread.sleep(1000);

            ob1.mysuspend();
            System.out.println("Suspending thread.");
            Thread.sleep(1000);

            ob1.myresume();
            System.out.println("Resuming thread.");
            Thread.sleep(1000);

            ob1.mysuspend();
            System.out.println("Suspending thread.");
            Thread.sleep(1000);

            ob1.myresume();
            System.out.println("Resuming thread.");
            Thread.sleep(1000);

            ob1.mysuspend();
            System.out.println("Stopping thread.");
            ob1.mystop();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }
    }
}

```

OUTPUT

```
$ javac Suspend.java && java Suspend  
My Thread Starting.
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40
```

```
Suspending thread.  
Resuming thread.
```

```
41  
42  
43  
44  
45  
46
```

47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70

71
72
73
74
75
76
77
78
79
80

Suspending thread.
Resuming thread.

81
82
83
84
85
86
87
88
89
90
91
92
93
94

```

95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120

```

Stopping thread.
My Thread exiting.

Exercise Questions - 3

Q. Write a program that, using an integer array of date values containing month, day, and year as integers for some number of dates (10, say, so the integer array will be two-dimensional with 10 rows and 3 columns), will write a file with a string representation of each date written as Unicode characters. For example, the date values 3,2,1990 would be written to the file as 2nd March 1990. Make sure that the date strings can be read back, either by using a separator character of some kind to mark the end of each string or by writing the length of each string before you write the string itself.

```

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

```

```
class Date {  
    private static final String[] months = { "January", "February", "March",  
    "April", "May", "June", "July", "August",  
    "September", "October", "November", "December" };  
  
    int dd, mm, yy;  
  
    Date(int dd, int mm, int yy) {  
        this.dd = dd;  
        this.mm = mm;  
        this.yy = yy;  
    }  
  
    private static String getDate(int dd) {  
        String suffix = "th";  
        if (dd == 1 || dd == 21 || dd == 31)  
            suffix = "st";  
        else if (dd == 2 || dd == 22)  
            suffix = "nd";  
        else if (dd == 3 || dd == 23)  
            suffix = "rd";  
        return dd + suffix;  
    }  
  
    private String getMonth(int mm) {  
        return months[mm - 1];  
    }  
  
    private static int getMonth(String month) {  
        switch (month) {  
            case "January":  
                return 1;  
            case "February":  
                return 2;  
            case "March":  
                return 3;  
            case "April":  
                return 4;  
            case "May":  
                return 5;  
            case "June":  
                return 6;  
            case "July":  
                return 7;  
            case "August":  
                return 8;  
            case "September":  
                return 9;  
            case "October":  
                return 10;  
            case "November":  
                return 11;  
        }  
    }  
}
```

```

        return 11;
    default:
        return 12;
    }
}

static Date getDate(String date) {
    String[] values = date.split("-");
    int dd = Integer.parseInt(values[0].substring(0, values[0].length() - 2));
    int mm = getMonth(values[1]);
    int yy = Integer.parseInt(values[2]);
    return new Date(dd, mm, yy);
}

public String toString() {
    return (getDate(this.dd) + "-" + getMonth(this.mm) + "-" + this.yy);
}
}

public class Exercise2 {
    public static void main(String[] args) throws IOException {
        int[][] dates = { { 21, 1, 1998 }, { 1, 2, 1999 }, { 12, 3, 2000 }, { 26, 4,
2001 }, { 13, 5, 2002 },
                { 2, 6, 2003 }, { 5, 7, 2004 }, { 10, 8, 2005 }, { 20, 9, 2006 },
                { 31, 10, 2007 }, };
        FileWriter writer = new FileWriter("dates.txt");

        for (int i = 0; i < dates.length; i++) {
            Date d = new Date(dates[i][0], dates[i][1], dates[i][2]);
            String date = d.toString();
            System.out.println("Written: " + date);
            writer.write(date + "\n");
        }

        System.out.println("\nReading Data: ");
        writer.close();

        BufferedReader reader = new BufferedReader(new
FileReader("dates.txt"));
        String date;

        while ((date = reader.readLine()) != null) {
            System.out.println("Read: " + Date.getDate(date));
        }

        reader.close();
    }
}

```

OUTPUT

```
$ javac Exercise2.java && java Exercise2
Written: 21st-January-1998
Written: 1st-February-1999
Written: 12th-March-2000
Written: 26th-April-2001
Written: 13th-May-2002
Written: 2nd-June-2003
Written: 5th-July-2004
Written: 10th-August-2005
Written: 20th-September-2006
Written: 31st-October-2007
```

Reading Data:

```
Read: 21st-January-1998
Read: 1st-February-1999
Read: 12th-March-2000
Read: 26th-April-2001
Read: 13th-May-2002
Read: 2nd-June-2003
Read: 5th-July-2004
Read: 10th-August-2005
Read: 20th-September-2006
Read: 31st-October-2007
```

Q. Extend the previous example to write a second file at the same time as the first, but containing the month, day, and year values as binary data. You should have both files open and be writing to both at the same time.

```
import java.io.*;

class Date {
    private static final String[] months = { "January", "February", "March",
    "April", "May", "June", "July", "August",
    "September", "October", "November", "December" };

    int dd, mm, yy;

    Date(int dd, int mm, int yy) {
        this.dd = dd;
        this.mm = mm;
        this.yy = yy;
    }

    private static String getDate(int dd) {
        String suffix = "th";
        if (dd == 1 || dd == 21 || dd == 31)
            suffix = "st";
```

```

        else if (dd == 2 || dd == 22)
            suffix = "nd";
        else if (dd == 3 || dd == 23)
            suffix = "rd";
        return dd + suffix;
    }

    private String getMonth(int mm) {
        return months[mm - 1];
    }

    private static int getMonth(String month) {
        switch (month) {
            case "January":
                return 1;
            case "February":
                return 2;
            case "March":
                return 3;
            case "April":
                return 4;
            case "May":
                return 5;
            case "June":
                return 6;
            case "July":
                return 7;
            case "August":
                return 8;
            case "September":
                return 9;
            case "October":
                return 10;
            case "November":
                return 11;
            default:
                return 12;
        }
    }

    static Date getDate(String date) {
        String[] values = date.split("-");
        int dd = Integer.parseInt(values[0].substring(0, values[0].length() - 2));
        int mm = getMonth(values[1]);
        int yy = Integer.parseInt(values[2]);
        return new Date(dd, mm, yy);
    }

    public String toString() {
        return (getDate(this.dd) + "-" + getMonth(this.mm) + "-" + this.yy);
    }
}

```

```

class Exercise3 {
    public static void main(String[] args) throws IOException {
        int[][] dates = { { 1, 1, 1999 }, { 1, 2, 1999 }, { 1, 3, 1999 }, { 1, 4,
1999 }, { 1, 5, 1999 },
        { 1, 6, 1999 }, { 1, 7, 1999 }, { 1, 8, 1999 }, { 1, 9, 1999 }, { 1,
10, 1999 }, };

        FileWriter writer = new FileWriter("dates.txt");
        DataOutputStream os = new DataOutputStream(new
FileOutputStream("binary.dat"));

        for (int i = 0; i < dates.length; i++) {
            Date d = new Date(dates[i][0], dates[i][1], dates[i][2]);
            os.writeInt(dates[i][0]);
            os.writeInt(dates[i][1]);
            os.writeInt(dates[i][2]);
            String date = d.toString();
            System.out.println(">>" + date);
            writer.write(date + "\n");
        }

        System.out.println("\nReading: ");
        writer.close();
        os.close();

        BufferedReader reader = new BufferedReader(new
FileReader("dates.txt"));
        DataInputStream is = new DataInputStream(new
FileInputStream("binary.dat"));
        String date;

        while ((date = reader.readLine()) != null) {
            System.out.println("Read: " + Date.getDate(date));
        }
        System.out.println("\nBinary file: \n ");

        int datesRead[][] = new int[20][];
        int j = 0;
        while (is.available() > 0) {
            int dd = is.readInt();
            int mm = is.readInt();
            int yy = is.readInt();
            datesRead[j] = new int[3];
            datesRead[j][0] = dd;
            datesRead[j][1] = mm;
            datesRead[j][2] = yy;
            j++;
        }

        System.out.print(dd + " - ");
        System.out.print(mm + " - ");
        System.out.println(yy);
    }
}

```

```
    }  
  
    reader.close();  
    is.close();  
}  
}
```

OUTPUT

>> 1st-January-1999
>> 1st-February-1999
>> 1st-March-1999
>> 1st-April-1999
>> 1st-May-1999
>> 1st-June-1999
>> 1st-July-1999
>> 1st-August-1999
>> 1st-September-1999
>> 1st-October-1999

Reading:

Read: 1st-January-1999
Read: 1st-February-1999
Read: 1st-March-1999
Read: 1st-April-1999
Read: 1st-May-1999
Read: 1st-June-1999
Read: 1st-July-1999
Read: 1st-August-1999
Read: 1st-September-1999
Read: 1st-October-1999

Binary file:

1 - 1 - 1999
1 - 2 - 1999
1 - 3 - 1999
1 - 4 - 1999
1 - 5 - 1999
1 - 6 - 1999
1 - 7 - 1999
1 - 8 - 1999
1 - 9 - 1999
1 - 10 - 1999

Q. Write a program that, for a given String object defined in the code, will write strings to a file in the local character encoding (as bytes) corresponding to all possible permutations of the words in

the string. For example, for the string the fat cat, you would write the strings the fat cat, the cat fat, cat the fat, cat fat the, fat the cat, and fat cat the, to the file, although not necessarily in that sequence. (Don't use very long strings; with n words in the string, the number of permutations is n!).

```

import java.io.*;

class A {
    public static void main(String[] args) throws IOException {
        String str = "ben ten den";
        String[] words = str.split(" [ ]");

        FileWriter writer = new FileWriter("permu.txt");

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                for (int k = 0; k < 3; k++) {
                    String data = words[i] + words[j] + words[k] + "\n";
                    writer.write(data);
                }
            }
        }

        writer.close();

        BufferedReader reader = new BufferedReader(new
FileReader("permu.txt"));

        while ((str = reader.readLine()) != null)
            System.out.println(str);

        reader.close();
    }
}

```

OUTPUT

```

benbenben
benbenten
benbenden
bentenben
bententen
bentenden
bendenben
bendeten
bendenden
tenbenben
tenbenten

```

tenbenden
tentenben
tententen
tentenden
tendenben
tendenten
tendenden
denbenben
denbenten
denbenden
dentenben
dententen
dentenden
dendenben
dendenten
dendenden

THE END