22/10/20. Model Exam M. Hari Haran

OOPS 19CSE042.

1) Ans

Yes, you can save your Java source file with any other name, not same as your main class name but when you compile it that byte code file name will be same as your main class name. If there is any public class in file then name as file name.

2) Java is a platform Independent because it is different from other languages like c, c++ etc, which are compiled into platfrom specific machines while java is a write one, run any where java language. A platfrom is the hardware or software environment in which a program runs.

3) <u>method overriding in Java</u>

* The argument list should be correctly the same as that ~~of~~ the overridden method.

* constructors cannot be overridden

* A method ~~de~~clared final cannot be overridden

5) <u>stack trace element</u>

* <u>string</u> getclass name ( )

This method returns the fully qualified name of the class execution point represented by this stack trace element

* <u>String getmethod</u> name ( )

This method returns the name of the method containing the execution point.

6) <u>Input stream :-</u>

⇒ It is present in byte stream

⇒ byte stream consists of image, video, audio

⇒ can read, byte at a time

7) <u>run method</u>

we need start (), method to call run(), run is only present in thread class. so we need start (), without start () we cannot call run ()

8) <u>Need for Generic code</u>

with a ~~type~~ variable we can access a specific code with many datatypes.

Scanned with CamScanner

9) import java . awt . Graphics ;
public class demo extends Applet
{
String s = " Hello world " ;
public void init () {}
public void paint (Graphics g)
{
g .draw String (s, 100, 200) ;
}
}

10) public class demo .
{
J ~~frame~~ frame f ;
demo)
{
f = new J frame ("demo 2") ;
f . set Size (600, 400) ;
f . set Layout (null) ;
f = set visible (true) ;
}
public static void main (string [] args)
{ new demo () ;
}
}

part B

11) (b) Static in java

Static can be used before a variable, method.

Static members

1) variables.

2) methods.

3) static blocks.

Static variable

memory is allocated only once for static variable.

Static method

when a method is declared startic there is no need to create an object we need to call the method using the class name where the static method is present.

static block

static block can be present any where in the program.

when every the static block is the body of the static block will be executed first even before the execution of main method.

// static method.

```
package aaa;
public .class bbb
{
 public state void main ( string [] args )
{
 ccc .display ( );
}
}
class ccc
{ static void display ( )
{
 system. out print/n ( " static method" );
}
}.
```

O/P.

```
static method
```

constructos

⇒ special member of class.

⇒ allocting memory while creating an obyect

1) Default constructas

2) paramantaized constructor

1) **Default constructor**

It there is no user defend constructor the program will create a default constructor.

2) **parmenterixed** constructor

```
class box
{
box (int a, int b)
}
```

⇒ name of the constructas should be same as the class name.

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

 {
  box obj = new box1 ();
  system . out . println (obj . depth );
  box1 obj1 = new box (10 , 20 , 30 );
  system . out . println (obj . depth );

 }
}

class box 1.
{
  double ht ;
  double width ;
  double depth ;
  box1 ()
 {
   depth = 10 ;
 }
 box1 ( double h , double w , double d )
 { ht = h ;
   width = w ;
   depth = d ;
 }
}
```

12) b)

```
import Java. util . scanner ;

public class Java example

{
  public static void main ( String args [] )

{
 int marks [] = new int [6] ;
 int i ;
 float total = 0 , avg ;
 scanner scanner = new scanner ( System . in ) ;

 for ( i = 0 ; i < 6 ; i++ )
 {
  system . out . print ( " Enter marks of subject " + ( i+1 )
                                                   + " : " ) ;
 marks [i] = scanner . nextint ( ) ;
  total = total + marks. [i] ;

}
 scanner . close ( ) ;
 avg = total / 6 ;
 system . out . print ( " the student Grade is : " ) ;
 if ( avg > = 80 )
{
  system . out . print ( " A " ) ;

}
else if (avg >= 60 && avg < 80)
{
system.out.print("B");
}
else if (avg >= 40 && avg < 60)
{
system.out.print("C");
}
else
{
system.out.print("D");
}
}
}
```

O/P

```
Enter marks of sub 1 : 40
   "           sub 2 : 80
   "           sub 3 : 80
   "           sub 4 : 40
   "           sub 5 : 60
   "           sub 6 : 66
The student grade is : B
```

13) A) Java nested :-

The try block within a try block is known as nested try block in Java.

use of nested

Sometime a situation may arise where a ~~so~~ part of a block may cause one error and the entire of block may cause another error. In such cases, exception handlers have to be nested.

Syntax : .

```
try
{
 Statement 1 ;
 Statement 2 ;
 try
 {
  statement 1 ;
  statement 2 ;
} try
{
```

```
catch (Exception e )

{

}

}

catch (Exception e )

{

}
```

Ex

13;

```
    class Excep6 {
    public static void main ( string args [] )
    {
      try
    {
    try {
    system . out. Println ( " going to divide " );
    int  b = 39/0 ;
    }
    catch (Arithmetic Exception e )
    { system . out . println (e); }

    try {
      int a [] = new int [5] ;
      a [5] = 4 ;
    }
```

```
catch
{
system.out.println(e); }
system.out.print~~B~~ln("handeled"); }
system.out.println("normal flow..");
}
}
```

13). A (ii) <u>user defined</u>

In java we have already defined, exception classes such as Arithmetic Exception null pointer Exception etc. these exception are already ~~set~~ to trigger on pre-defined condition sets to trigger on [redacted] - when you divide a number by zero it triggers Arithmetic Exception, In the last tutorial we learnt how to you throw these exception

To understand this turtorial you should have basic knowledge of try-catch block and throw in Java.

Ex

```
class My Exception extends Exception
{
  strings str 1;
  My Exception (string str 2)
  {
   str 1 = Str 2
  }
  public string to string ()
  {
   return ("My Exception occurred: " + str 1);
  }
}
class Example 1 {
public static void main (string args[])
{
  try
   {
     system.out.print/n ("starting of try
                              block");
  throw new My Exception ("This is my error
                              Message");
  }
catch (My Exception exp)
{
system . out . println ("catch Block");
system . out . println (exp);
}
}
}
```

O/P

```
Starting of try block
catch Block
My Exception occured: This is My error Msg
```

Scanned with CamScanner

14) a) Ans

Inter thread communication in Java

It is all about allowing synchronized threads to communicate with each other.

It is a mechanism in which at thread is paused running is it critical such and another. thread is allowed to enter in the same critical section to be executed.

Object class

* wait ( )
* notify ( )
* notify All ( ).

Suspending resuming and stoping thread the following example illustrate how to write and notify ( ) method that are inheritable from object. It can be used to control the excution of thread. the new thread class contains a boolean instance variable names suspend flag.

Scanned with CamScanner

```
class new thread implements Runable
{
string name ;
thread ;
boolean suspend flag ;
new thread ( string thread name ) ;
{
name = thread name ;
L = new thread ( this, name ) ;
system . out . print ln ("new thread:" +1 ) ;
suspend flag = false ;
L start () ;
}
public void run () {
try {
fort ( int i = 15 ; i > 0 ; i -- )
{
~~public ( void run () ~~ {
system . out . println ( name + " ; " +1 ) ;
thread . slep ( 200 ) ;
synchronized ( thus )
{
```

```
while (suspend flag )
 { wait ();
 }
 }
}
}
catch (Interupted exception)
 {
 system .out .println (name + " Intesupeted ");
 }
 systemout . println (name + "existing " );
 }
 synchronize void mysuspend ()
 {
 suspend flag = true ;
 }
 synchronised void and rasume ( )
 {
 suspend flag = false ;
 notify () ;
 }
}
```

Scanned with CamScanner

```
public class suspend Resume
{
 public static void main ( string args [] )

{  new thread of 1 = new new / thread (" one");
   new thread of 2 = new new / thread (" Two");

  try
{
   Thread sleep (1000 );
   of 1. my suspend ( );
   system . out . println (" suspend thread one");
   thread . sleep ( 1000);
   of 1. myresume ( );
   system . out . println (" suspend thread Two");
   Thread . sleep ( 1000 );
   of 2 my  resume ( );
   system . out . println (" Resume  thread two ");

 }.
   catch ( Interrupted . Exception )
{
 system . out . println ("Main thread
                                Intersuspend");
 }
 try
  {
   system  .out. println ("waiting for
                          thread to finish.");
   of 1 . t . join ();
   of 2 . t . join ();
  }
   catch (interspent exception)
  {
   system  .out  . print ln ("Main thread Intesupend");
 }
 system  . out  . print ln ("Main thread
                          existing.");
 }
}.
```

//

15) (a)

```
Import java . applet . Applet ;
import java . awt. * ;
import java . awt . event. * ;

/*
 * <applet code =
" Traffic lights Example" width =1000
  height = 500 >
 * </applet>
 ** /

public class Traffic lights Example
extends Applet implements
Item Listener
{
  check box group grp = new
 check box group ( ) ;
 check box red light , yellow Light
green light ;
   Label msg ;
   public void init ( )
{ red light = new checkbox ("Red", grp, false);
```

Scanned with CamScanner

```
yellowLight = new
checkbox ("yellow", grb, false);
 green light = new
checkbox ("Green", grp, false);
  msg = new Label ("");
redlight.additemListener (this);
yellow light.additem Listener (this);
green light.additem listener (this);
    add (redlight);
    add (yellowlight);
    add (green light);
    add (msg);
 Msg. setfont (new font ("serif", font.BOLD, 20));
}
 public void
 item state change (item Event ie)
 { redlight. setforeground (Color.BLACK);
   yellowlight. setforeground (color.BLACK);
   green light. setforeground (color.BLACK);
```

```
if (redlight.getState() == true)
{
redlight.setforeground(color.RED);
  msg.setforground(color.RED);
  msg.setText("stop");
 }
 else if (yellowlight.getstate() = true)
 {
 yellowlight.setforeground(color.YELLOW);
   msg.setforground (color.YELLOW);
   msg.Set Text ("~~STOP~~ READY");
  }
  else if  (GREEN light.get state() = true)
  {
   greenlight.setforeground (color.GREEN);
  msg.setforground (color.GREEN
  msg.setText ( "GO");
 }
}
}
   //
```