MOHAMED SATHAK A J COLLGE OF ENGINEERING, Chennai – 603103

Department of Electrical And Electronics Engineering Year: III

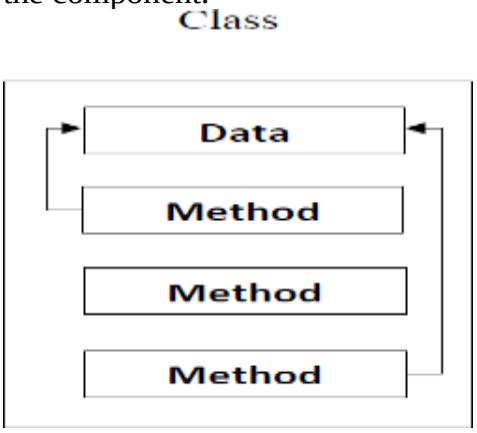
CS8392 - OBJECT ORIENTED PROGRAMMING

UNIT I INTRODUCTION TO OOP AND JAVA FUNDAMENTALS

1.Define Objects and Classes in Java(Nov/Dec 2018)

Class is a collection of data and the function that manipulate the data.The data components of the class are called data fields and the function components of the class are called member functions or methods. The class that contains main function is called main class.
Object is an instance of a class. The objects represent real world entity. The objects are used to provide a practical basis for the real world. Objects are used to understand the real world. The object can be declared by specifying the name of the class.

2.Define Encapsulation (Apr/May 2012) (Apr 2017)
The wrapping up of data and functions into a single unit is known as data encapsulation. Here the data is not accessible to the outside the class. The data inside that class is accessible by the function in the same class. It is normally not accessible from the outside of the component.



3.List any four Java Doc comments. [NOV/DEC 2011]
A Javadoc comment is set off from code by standard multi-line comment tags /* and */. The opening tag, however, has an extra asterisk, as in /**.The first paragraph is a description of the method documented. Following the description are a varying number of descriptive tags, signifying: The parameters of the method (@param),What the method returns (@return) and any exceptions the method may throw (@throws).

4.What is the access specifier?9 (Nov Dec 2019)
Java Access Specifiers (also known as Visibility Specifiers) regulate access to classes, fields and methods in java. These specifiers determine whether a field or method in c lass, can be used or invoked by another method in another class or sub-class. Access Specifiers can be used to restrict access. There are 4 types of java access modifiers: Private, Default, Protected

and Public.

5.Define Abstraction. (May / June 2016)

Abstraction refers to the act of representing the essential features without including the background details or explanations. It reduces the complexity and increases the efficiency. Smallprograms can be easily upgraded to large programs. Software complexity can easily be managed.

6. What is Polymorphism?

Polymorphism is the ability to take more than one form and refers to an operation exhibiting different behavior instances. Object oriented programs use polymorphism to carry out the sameoperation in a manner customized to the object. It allows a single name/operator to be associated with different operation depending on the type of data passed to it.

7. What is Inheritance? What are its types?

Inheritance is a mechanism of reusing the properties and extending existing classes without modifying them, thus producing hierarchical relationships between them.

Inheritance is a property by which the new classes are created using the old classes.

The old classes are referred as base classes and the new classes are referred as derivedclasses. That means the derived classes inherit the properties of base class.

extends and implements keywords are used to describe inheritance in Java.

Types of inheritance are: Single inheritance, Multi-level inheritance, Hierarchical inheritance, Hybrid inheritance.

Syntax :

```
class Subclass-name extends Superclass-name
{       //methods and fields   }
```

8. Define class[NOV/DEC 2011]

Class is a template for a set of objects that share a common structure and a common behavior.

9. What is method in java? How to define and call the method?

Method is a programming construct used for grouping the statement together to build afunction. There are two ways by which the method is handled.

1. Defining a method 2. Calling a method

Here is example that helps to understand the concept of method defining and calling. public class methDemo {

```
public static void main(String args[]) {int a=10;int b=20;int c=sum(a,b);
System.out.println("The sum of "+a"+" and "+b+" is="+c);
}
public static int sum(int num1,int num2)
{
int ans=num1+num2;return ans;}}
```

10. What are the control flow statements in java?

A programming language uses control statements to control the flow of execution of program

based on certain conditions. These are used to cause the flow of execution to advance and branch based on changes to the state of a program.
Java's Selection statements:
if
if-else
nested-if
if-else-if
switch-case
jump – break, continue, return
These statements allow you to control the flow of your program's execution based upon conditions known only during run time.

11. What do you mean by Variable? What are the rules for variable declaration?

Variable is a fundamental unit of storage in java. The variables are used in combination with identifiers, data types, operators and some value for initialization.
The syntax of variable declaration will be: data_type name_of_variable[=initialization];

12. What do you mean by Bytecode? What is JVM and JIT?

Bytecode is an intermediate form of java programs.We get bytecode after compiling the java program using a compiler called javac. The bytecode is to be executed by java runtime environment which is called as Java Virtual Machine(JVM). The programs that are running on JVM must be compiled into a binary format which is denoted by .class files. The JVM executes
.class or .jar files, by either interpreting it or using a just-in-time compiler (JIT). The JIT is used for compiling and not for interpreting the file. It is used in most JVMs today to achieve greater
speed.

UNIT II INHERITANCE AND INTERFACES

1.What is meant by Inheritance and what are its advantages?

Inheritance is a relationship among classes, wherein one class shares the structure or behavior defined in another class. This is called Single Inheritance. If a class shares the structure or behavior from multiple classes, then it is called Multiple Inheritance. Inheritance defines "is-a" hierarchy among classes in which one subclass inherits from one or more generalized super
classes. The advantages of inheritance are reusability of code and accessibility of variables andmethods of the super class by subclasses.

2.What is final modifier?

The final modifier keyword makes that the programmer cannot change the value anymore. Theactual meaning depends on whether it is applied to a class, a variable, or a method.
final Classes- A final class cannot have subclasses.
final Variables- A final variable cannot be changed once it is initialized. final Methods- A final method cannot be overridden by subclasses.

3.What is extending interface?

An interface can extend another interface in the same way that a class can extend another class. The extends keyword is used to extend an interface, and the child interface inherits the methods of the parent interface.

Syntax: interface interface_name{ Public void method1():
Public void method2(): }

4.What is object cloning?

It is the process of duplicating an object so that two identical objects will exist in the memory at the same time.

5.Define Array list class.

The ArrayList class extends AbstractList and implements the List interface. ArrayList is a generic class that has this declaration:

class ArrayList<E>

Here, E specifies the type of objects that the list will hold. An ArrayList is a variable-length array of object references. That is, an ArrayList can dynamically increase or decrease in size. Array lists are created with an initial size. When this size is exceeded, the collection is automatically enlarged. When objects are removed, the array can be shrunk.

6.What is String in Java? Is String is data type?

String in Java is not a primitive data type like int, long or double. String is a class or in more simple term a user defined type. String is defined in java.lang package and wrappers its content in a character array. String provides equals() method to compare two String and provides various other method to operate on String like toUpperCase() to convert String into upper case, replace() to replace String contents, substring() to get substring, split() to split long String into multiple String.

7.What is meant by Binding, Static binding, Dynamic binding?

Binding: Binding denotes association of a name with a class. *Static binding:* Static binding is a binding in which the class association is made during compile time. This is also called as *Early* binding. *Dynamic binding*: Dynamic binding is a binding in which the class association is not made until the object is created at execution time. It is also called as *Late* binding.

8.Brief Inner class in Java with its syntax.

Java inner class or nested class is a class which is declared inside the class or interface.

We use inner classes to logically group classes and interfaces in one place so that it can be more readable and maintainable.

Additionally, it can access all the members of outer class including private data members and methods.

Syntax of Inner class

```
class Java_Outer_class{
 //code
 class Java_Inner_class{
```

```
        //code
        }
}
```

9.What is the difference between abstract class and interface?

ABSTRACT CLASS	INTERFACE
1. Abstract class must have at least one abstract method and others may be concrete or abstract	All the methods declared inside an interface are abstract
2. In abstract class, key word abstract must be used for the methods	Interface we need not use that keyword for the methods.
3. Abstract class must have subclasses	Interface can't have subclasses

10.What are the methods provided by the object class?

The Object class provides five methods that are critical when writing multithreaded Java programs:
notify
notify All
wait (three versions)

11. What are the four types of access modifiers?

There are 4 types of java access modifiers:

1. private
2. default
3. protected
4. public

12. What are inner class and anonymous class?

Inner class: classes defined in other classes, including those defined in methods are called inner classes. An inner class can have any accessibility including private. Anonymous class: Anonymous class is a class defined inside a method without a name and is instantiated and
declared in the same place and cannot have explicit constructors

UNIT III Exception Handling and I/O

1.Define Java Exception.

A Java exception is an object that describes an exceptional (that is, error) condition that has occurred in a piece of code. When an exceptional condition arises, an object representing that

exception is created and thrown in the method that caused the error.

2.Name any four java built in exceptions.

Exception	Meaning
ArithmeticException	Arithmetic error, such as divide-by-zero.
ArrayIndexOutOfBoundsException	Arithmetic Exception Array index is out-of-bounds.
ArrayStoreException	Assignment to an array element of an incompatible type.
ClassCastException	Invalid cast

3. What is chained exception?

Chained Exceptions allows to relate one exception with another exception, i.e one exception describes cause of another exception. For example, consider a situation in which a method throws an ArithmeticException because of an attempt to divide by zero but the actual cause exception was an I/O error which caused the divisor to be zero.

4. What does java.lang.StackTraceElement represent?

The java.lang.StackTraceElement class element represents a single stack frame. All stack frames
except for the one at the top of the stack represent a method invocation. The frame at the top ofthe stack represents the execution point at which the stack trace was generated.

5. Compare throw and throws.

Throw is used to throw an exception & throws is used to declare an exception.
Throw is used in method implementation & throws is used in method signature.
Using throw keywordwe can throw only 1 exception at a time & throws can declare multiple exceptions at a time.

6.What is difference between final, finally and finalize in Java?

Final and finally are keywords in java whereas finalize is a method.
Final keyword can be used with class variables so that they can't be reassigned, with class to avoid extending by classes and with methods to avoid overriding by subclasses.
Finally keyword is used with try-catch block to provide statements that will always gets executed even if some exception arises, usually finally is used to close resources.
finalize() method is executed by Garbage Collector before the object is destroyed, it's great way to make sure all the global resources are closed. Out of the three, only finally is related to java exception handling.

7. What are input and output streams?

An I/O Stream represents an input source or an output destination. A stream can represent many different kinds of sources and destinations, including disk files, devices, other programs,and memory arrays.

8. What is the use of java console class?

The Java Console class is be used to get input from console. It provides methods to read texts and passwords. If you read password using Console class, it will not be displayed to the user.The java.io.Console class is attached with system console internally.

9. Can we have an empty catch block?
We can have an empty catch block but it's the example of worst programming. We should never have empty catch block because if the exception is caught by that block, we will have noinformation about the exception and it will be a nightmare to debug it.

10. What is the use of finally exception?
Finally block is optional and can be used only with try-catch block. Since exception halts the process of execution, we might have some resources open that will not get closed, so we can usefinally block. finally block gets executed always, whether exception occurs or not.

11. How Java Exception Hierarchy categorized?
Java Exceptions are hierarchical and inheritance is used to categorize different types of exceptions. Throwable is the parent class of Java Exceptions Hierarchy and it has two child objects – Error and Exception. Exceptions are further divided into checked exceptions and
runtime exception.

12. What is OutOfMemoryError in Java?
OutOfMemoryError in Java is a subclass of java.lang.VirtualMachineError and it's thrown byJVM when it ran out of heap memory.

UNIT IV Exception Handling and I/O

1.Define Thread?
A thread is a single sequential flow of control within program. Sometimes, it is called an execution context or light weight process. A thread itself is not a program. A thread cannot run on its own. Rather, it runs within a program. A program can be divided
into a number of packets of code, each representing a thread having its own separate flow of control.

2.Difference between multi-threading and multi-tasking?

Multi-threading	Multi-tasking
In any single process, multiple threads is allowed and again, can run simultaneously.	It refers to having multiple (programs, processes, tasks, threads) running at the same time.
It is sharing of computing resources among threads of a single process.	It is sharing of computing resources(CPU, memory, devices, etc.) among processes

3.What do you mean by Thread Scheduling?
Execution of multiple threads on a single CPU in some order is called scheduling. The Java

runtime environment supports a very simple, deterministic scheduling algorithm called fixed-priority scheduling. This algorithm schedules threads on the basis of their priority relative to other Runnable threads.

4. What is Thread Pool?

A thread pool is a managed collection of threads that are available to perform tasks. Thread pools usually provide:

Improved performance when executing large numbers of tasks due to reduced per-task invocation overhead

A means of bounding the resources, including threads, consumed when executing a collection of tasks.

5. What is thread priority?

Every Java thread has a priority that helps the operating system determine the order in which threads are scheduled. Java priorities are in the range between MIN_PRIORITY(a constant of 1) and MAX_PRIORITY(a constant of 10). By default, every thread is given priority NORM_PRIORITY(a constant of 5)Threads with higher priority are more important to a program and should be allocated processor time before lower-priority threads. However, thread priorities cannot guarantee the order in which threads execute and very much platform independent.

6. List out the methods of object class to perform inter thread communication?

wait() – This method make the current thread to wait until another thread invokesthe notify() method.

notify() – This method wakes up a thread that called wait() on same object.

notifyAll() – This method wakes up all the thread that called wait() on same object.Wakes up all threads that are waiting on this object's monitor.

Above all three methods have been implemented as final method in Object class, so that they are available in all the classes in java world.

7. Why do we need run() and start() method both? Can we achieve it with only run method?

The separate start() and run() methods in the Thread class provide two ways to create threaded programs. The start() method starts the execution of the new thread and calls the run() method. The start() method returns immediately and the new thread normally continues until the run() method returns.

The Thread class' run() method does nothing, so sub-classes should override the method with code to execute in the second thread. If a Thread is instantiated with a Runnable argument, the thread's run() method executes the run() method of the Runnable object in the new thread instead.

Depending on the nature of your threaded program, calling the Thread run() method directly can give the same output as calling via the start() method, but in the latter case the code is actually executed in a new thread.

8. Write short note on isAlive() and join()?

isAlive() and join() methods are used to determine whether a thread has finished or not.

First, you can call isAlive() on the thread. This method is defined by Thread, and its general form is:
final Boolean isAlive()
The isAlive() method returns true if the thread upon which it is called is still running. It returnsfalse otherwise.
While isAlive() is occasionally useful, the method that you will more commonly use towait for a thread to finish is called join(). The general form is:
final void join() throws InterruptedException
This method waits until the thread on which it is called terminates.

9. Define thread group?
Every Java thread is a member of a thread group. Thread groups provide a mechanism for collecting multiple threads into a single object and manipulating those threads all at once, rather than individually. For example, you can start or suspend all the threads within a group with a single method call.

10. What are the restrictions on generics?
To use Java generics effectively, you must consider the following restrictions:
Cannot Instantiate Generic Types with Primitive Types
Cannot Create Instances of Type Parameters
Cannot Declare Static Fields Whose Types are Type Parameters
Cannot Use Casts or instance of With Parameterized Types
Cannot Create Arrays of Parameterized Types
Cannot Create, Catch, or Throw Objects of Parameterized Types
Cannot Overload a Method Where the Formal Parameter Types of Each Overload Erase to the Same Raw Type

11. Define Deadlock and When it will occur?
Deadlock describes a situation where two or more threads are blocked forever, waiting for each other. Deadlock occurs when multiple threads need the same locks but obtain them in different order. A Java multithreaded program may suffer from the deadlock condition because the synchronized keyword causes the executing thread to block while waiting for the lock, ormonitor, associated with the specified object**.**

12. How to create generic class?
A class that can refer to any type is known as generic class. Here, we are using **T** typeparameter to create the generic class of specific type.
Let's see the simple example to create and use the generic class.
Creating generic class:

```
class
MyGen<T>{ T
obj;
void                    add(T
obj){this.obj=obj;}        T
get(){return obj;} }
```

The T type indicates that it can refer to any type (like String, Integer, Employee etc.). The

type
you specify for the class, will be used to store and retrieve the data.

UNIT V EVENT DRIVEN PROGRAMMING

1. What are methods available in the Applet class?
init() - To initialize the applet each time it's loaded (or reloaded).
start() - To start the applet's execution, such as when the applet's loaded or when the user revisits a page that contains the applet.
stop() - To stop the applet's execution, such as when the user leaves the applet's page or quits the browser.
paint()- To display the image
destroy - To perform a final cleanup in preparation for unloading.

2. What is AWT?
A collection of graphical user interface (GUI) components that were implemented using native- platform versions of the components. These components provide that subset of functionality which is common to all native platforms. Largely supplanted by the Project Swing component
set.

3.What is the relationship between an event- listener interface and an event adapter class?
An event-listener interface allows describing the methods which must be implemented by one of the event handler for a specific event.
An event-adapter allows default implementations of an event-listener interface of a specific event.

4. List out some UI components available in AWT?
- Buttons (java.awt.Button)
- Checkboxes (java.awt.Checkbox)
- Single-line text fields (java.awt.TextField)
- Larger text display and editing areas (java.awt.TextArea)
- Labels (java.awt.Label)
- Lists (java.awt.List)
- Pop-up lists of choices (java.awt.Choice)
- Sliders and scrollbars (java.awt.Scrollbar)
- Drawing areas (java.awt.Canvas)
- Menus (java.awt.Menu, java.awt.MenuItem, java.awt.CheckboxMenuItem)
- Containers (java.awt.Panel, java.awt.Window and its subclasses)

5. How can you prevent the overwriting of a displayed text in a TextField of a java

program?
If you create a TextField object with default text then setting the prompt text will not overwritethe default text.
To set the prompt text for a TextField use the setPromptText method:
txtFld.setPromptText("Enter Name..");
To find out the value of the prompt text of a TextField object use the getPromptText method:
String promptext = txtFld.getPromptText();

6. How does a radio button in java differ from a check box?
Radio buttons are used when there is a list of two or more options that are mutually exclusive and the user must select exactly one choice. In other words, clicking a non-selected radio button will deselect whatever other button was previously selected in the list.
Checkboxes are used when there are lists of options and the user may select any number of choices, including zero, one, or several. In other words, each checkbox is independent of all other checkboxes in the list, so checking one box doesn't uncheck the others.

7. Name the listener methods that must be implemented for the key listener interface. (NOV2013)
void keyTyped(KeyEvent e)
void keyPressed(KeyEvent e)
void keyReleased(KeyEvent e)

8. Components of Event Handling
Event handling has three main components,
Events : An event is a change in state of an object.
Events Source : Event source is an object that generates an event.
Listeners : A listener is an object that listens to the event. A listener gets notified when an event occurs.

9. Mention the Differences between AWT and swing

Java AWT	Java Swing
AWT components are platform-dependent.	Java swing components are platform-independent.
AWT components are heavyweight.	Swing components are lightweight.
AWT provides less components than Swing.	Swing provides more powerful components such as tables, lists, scrollpanes, colorchooser,tabbedpane etc.
AWT doesn't follows MVC	Swing follows MVC.

10. What is a layout manager and what are different types of layout managers available in javaAWT?
A layout manager is an object that is used to organize components in a container. The different layouts are available are FlowLayout, BorderLayout, CardLayout, GridLayout and GridBagLayout.

11. What is the difference between choice and list?

A Choice is displayed in a compact form that requires you to pull it down to see the list of available choices and only one item may be selected from a choice. A List may be displayed in such a way that several list items are visible and it supports the selection of one or more list items.

12. How Events are handled in java ?

A source generates an Event and send it to one or more listeners registered with the source. Once event is received by the listener, they process the event and then return. Events are supported by a number of Java packages, like java.util, java.awt and java.awt.event.

MOHAMED SATHAK A J COLLGE OF ENGINEERING, Chennai – 603103

Department of Electrical And Electronics Engineering

CS8392 - OBJECT ORIENTED PROGRAMMING

UNIT I Introduction To oops and java Fundamentals

1. Explain the various features of the Object Oriented Programming Language
2. What are the different data types in JAVA? Explain each of them with example.
3. How to pass and return the objects to and from the method?
4. Discuss in detail the access specifiers available in Java.
5. Explain Packages in detail.
6. Explain Constructors with examples.
7. Explain in detail the various operators in Java.
8. Explain the concepts of arrays in Java and explain its types with examples?
9. Explain in detail about static variable and static method in Java with example?

UNIT II Inheritance and Interfaces

1. Explain the concept of inheritance with suitable examples.
2. Explain arrays in java with suitable example.
3. Explain how inner classes and anonymous classes works in java program.
4. What is a Package? What are the benefits of using packages? Write down the steps in creating apackage and using it in a java program with an example.
5. What is meant by object cloning? Explain it with an example.
6. Discuss in detail about inner class. With its advantages.
7. Explain about the object and abstract classes with the syntax.
8. Differentiate method overloading and method overriding. Explain both with an exampleprogram.
9. Explain interfaces with example.
10. Explain the concept of inheritance with suitable examples.
11. Explain arrays in java with suitable example.

UNIT III Exception Handling and I/O

1. Explain in detail the important methods of Java Exception Class?
2. Explain the different scenarios causing “Exception in thread main”?
3. How will you create your Own Exception Subclasses?
4. Explain in detail Chained exception with an example program.
5. Write programs to illustrate arithmetic exception, ArrayIndexOutOfBounds Exception and NumberFormat Exception.
6. Write a calculator program using exceptions and functions.
7. Create two exception classes that can be used by the stack classes developed by TRY

UNIT IV Exception Handling and I/O

1. What are the two ways of thread creation? Explain with suitable examples.
2. With illustrations explain multithreading, interrupting threads, thread states and thread properties.
3. Describe the life cycle of thread and various thread methods.
4. Explain the thread properties in detail.
5. Explain inter thread communication and suspending, resuming and stopping threads.
6. Write a java program for inventory problem to illustrate the usage of thread synchronized keyword and inter thread communication process. They have three classes called consumer,producer and stock.

UNIT V Event Driven Programming

1. What is event delegation model and what are the event classes and event interfaces?
2. Explain various components in AWT?
3. What is event handling in java? List out the available event classes and listener interfaces withsuitable example.
4. Explain the layout managers in Java also describe the concept of menu creation.
5. What is an adapter class? Describe about various adapter classes in detail?
6. Develop a java code that keeps the count of right clicks of mouse.
7. Explain about JButtonclass, JTextAreaclass, JFrameclass