DEPARTMENT OF III M.Sc., [Software Engineering]

INTERNET PROGRAMMING

2 MARKS

1. How could Java classes direct program messages to the system console, but error messages, say to a file?

The class System has a variable out that represents the standard output, and the variable err that represents the standard error device. By default, they both point at the system console. This how the standard output could be re-directed:

Stream st = new Stream(new FileOutputStream("output.txt")); System.setErr(st); System.setOut(st);

2. What's the difference between an interface and an abstract class?

An abstract class may contain code in method bodies, which is not allowed in an interface. With abstract classes, you have to inherit your class from it and Java does not allow multiple inheritance. On the other hand, you can implement multiple interfaces in your class.

3. Why would you use a synchronized block vs. synchronized method?

Synchronized blocks place locks for shorter periods than synchronized methods.

4. Explain the usage of the keyword transient?

This keyword indicates that the value of this member variable does not have to be serialized with the object. When the class will be de-serialized, this variable will be initialized with a default value of its data type (i.e. zero for integers).

5. How can you force garbage collection?

You can't force GC, but could request it by calling System.gc(). JVM does not guarantee that GC will be started immediately.

6. How do you know if an explicit object casting is needed?

If you assign a superclass object to a variable of a subclass's data type, you need to do explicit casting. For example: Object a; Customer b; b = (Customer) a; When you assign a subclass to a variable having a supeclass type, the casting is performed automatically.

7. What's the difference between the methods sleep() and wait()

The code sleep(1000); puts thread aside for exactly one second. The code wait(1000), causes a wait of up to one second. A thread could stop waiting earlier if it receives the notify() or

notifyAll() call. The method wait() is defined in the class Object and the method sleep() is defined in the class Thread.

8. Can you write a Java class that could be used both as an applet as well as an application?

Yes. Add a main() method to the applet.

9. What's the difference between constructors and other methods?

Constructors must have the same name as the class and can not return a value. They are only called once while regular methods could be called many times.

10. What is meant by Object Oriented Programming?

OOP is a method of programming in which programs are organised as cooperative collections of objects. Each object is an instance of a class and each class belong to a hierarchy.

11.What is a Class?

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

12.What is an Object?

Object is an instance of a class. It has state,behaviour and identity. It is also called as an instance of a class.

13.What is an Instance?

An instance has state, behaviour and identity. The structure and behaviour of similar classes are defined in their common class. An instance is also called as an object.

14.What are the core OOP's concepts?

Abstraction, Encapsulation,Inheritance and Polymorphism are the core OOP's concepts.

15.What is meant by abstraction?

Abstraction defines the essential characteristics of an object that distinguish it from all other kinds of objects. Abstraction provides crisply-defined conceptual boundaries relative to the perspective of the viewer. Its the process of focussing on the essential characteristics of an object. Abstraction is one of the fundamental elements of the object model.

16.What is meant by Encapsulation?

Encapsulation is the process of compartmentalising the elements of an abtraction that defines the structure and behaviour. Encapsulation helps to separate the contractual interface of an abstraction and implementation.

17.What are Encapsulation, Inheritance and Polymorphism?

Encapsulation is the mechanism that binds together code and data it manipulates and keeps both safe from outside interference and misuse. Inheritance is the process by which one object acquires the properties of another object. Polymorphism is the feature that allows one interface to be used for general class actions.

18.What are methods and how are they defined?

Methods are functions that operate on instances of classes in which they are defined. Objects can communicate with each other using methods and can call methods in other classes. Method definition has four parts. They are name of the method, type of object or primitive type the method returns, a list of parameters and the body of the method. A method's signature is a combination of the first three parts mentioned above.

19.What are different types of access modifiers (Access specifiers)?

Access specifiers are keywords that determine the type of access to the member of a class. These keywords are for allowingprivileges to parts of a program such as functions and variables. These are:

public: Any thing declared as public can be accessed from anywhere.

private: Any thing declared as private can't be seen outside of its class.

protected: Any thing declared as protected can be accessed by classes in the same package and subclasses in the other packages.

default modifier : Can be accessed only to classes in the same package.

20.What is an Object and how do you allocate memory to it?

Object is an instance of a class and it is a software unit that combines a structured set of data with a set of operations for inspecting and manipulating that data. When an object is created using new operator, memory is allocated to it.

21.Explain the usage of Java packages.

This is a way to organize files when a project consists of multiple modules. It also helps resolve naming conflicts when different packages have classes with the same names. Packages access level also allows you to protect data from being used by the non-authorized classes.

22.What is method overloading and method overriding?

When a method in a class having the same method name with different arguments is said to be method overloading. Method overriding : When a method in a class having the same method name with same arguments is said to be method overriding.

23.What gives java it's "write once and run anywhere" nature?

All Java programs are compiled into class files that contain bytecodes. These byte codes can be run in any platform and hence java is said to be platform independent.

24.What is a constructor? What is a destructor?

Constructor is an operation that creates an object and/or initialises its state. Destructor is an operation that frees the state of an object and/or destroys the object itself. In Java, there is no concept of destructors. Its taken care by the JVM.

25.What is the difference between constructor and method?

Constructor will be automatically invoked when an object is created whereas method has to be called explicitly

26.What is Static member classes?

A static member class is a static member of a class. Like any other static method, a static member class has access to all static methods of the parent, or top-level, class.

27.What is Garbage Collection and how to call it explicitly?

When an object is no longer referred to by any variable, java automatically reclaims memory used by that object. This is known as garbage collection. System. gc() method may be used to call it explicitly

28. In Java, How to make an object completely encapsulated?

All the instance variables should be declared as private and public getter and setter methods should be provided for accessing the instance variables.

29.What is static variable and static method?

static variable is a class variable which value remains constant for the entire class static method is the one which can be called with the class itself and can hold only the staic variables

30.What is finalize() method?

finalize () method is used just before an object is destroyed and can be called just prior to garbage collection.

31.What is the difference between String and String Buffer?

a) String objects are constants and immutable whereas StringBuffer objects are not.

b) String class supports constant strings whereas StringBuffer class supports growable and modifiable strings.

32.What is the difference between Array and vector?

Array is a set of related data type and static whereas vector is a growable array of objects and dynamic

33. What is the difference between this() and super()?

this() can be used to invoke a constructor of the same class whereas super() can be used to invoke a super class constructor.

34.Explain working of Java Virtual Machine (JVM)?

JVM is an abstract computing machine like any other real computing machine which first converts .java file into .class file by using Compiler (.class is nothing but byte code file.) and Interpreter reads byte codes.

35.How can you minimize the need of garbage collection and make the memory use more effective?

Use object pooling and weak object references.

36.There are two classes: A and B. The class B need to inform a class A when some important event has happened. What Java technique would you use to implement it?

If these classes are threads I'd consider notify() or notifyAll(). For regular classes you can use the Observer interface.

37.What access level do you need to specify in the class declaration to ensure that only classes from the same directory can access it?

You do not need to specify any access level, and Java will use a default package access

level.

38. What is garbage collection? What is the process that is responsible for doing that in java?

Reclaiming the unused memory by the invalid objects. Garbage collector is responsible for this process

39.What is a daemon thread?

These are the threads which can run without user intervention. The JVM can exit when there are daemon thread by killing them abruptly.

40.What is the finalize method do?

Before the invalid objects get garbage collected, the JVM give the user a chance to clean up some resources before it got garbage collected.

41. What is mutable object and immutable object?

If a object value is changeable then we can call it as Mutable object. (Ex., StringBuffer, …) If you are not allowed to change the value of an object, it is immutable object. (Ex., String, Integer, Float, …)

42. What is the basic difference between string and stringbuffer object?

String is an immutable object. StringBuffer is a mutable object.

43.What is the purpose of Void class?

The Void class is an uninstantiable placeholder class to hold a reference to the Class object representing the primitive Java type void.

44.What is an Abstract Class?

Abstract class is a class that has no instances. An abstract class is written with the expectation that its concrete subclasses will add to its structure and behaviour, typically by implementing its abstract operations.

45.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

46.What is an Interface?

Interface is an outside view of a class or object which emphaizes its abstraction while hiding its structure and secrets of its behaviour.

47.What is a base class?

Base class is the most generalised class in a class structure. Most applications have such root classes. In Java, Object is the base class for all classes.

48.What is reflection in java?

Reflection allows Java code to discover information about the fields, methods and constructors of loaded classes and to dynamically invoke them.

49.What is the difference between a static and a non-static inner class?

A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

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

a) All the methods declared inside an interface are abstract whereas abstract class must have at least one abstract method and others may be concrete or abstract.

b) In abstract class, key word abstract must be used for the methods whereas interface we need not use that keyword for the methods.

c) Abstract class must have subclasses whereas interface can’t have subclasses.

51.Can you have an inner class inside a method and what variables can you access?

Yes, we can have an inner class inside a method and final variables can be accessed.

52.What is interface and its use?

Interface is similar to a class which may contain method's signature only but not bodies and it is a formal set of method and constant declarations that must be defined by the class that implements it.

Interfaces are useful for:

a) Declaring methods that one or more classes are expected to implement

b) Capturing similarities between unrelated classes without forcing a class relationship. c) Determining an object's programming interface without revealing the actual body of the class.

53.What is a cloneable interface and how many methods does it contain?

It is not having any method because it is a TAGGED or MARKER interface.

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

The Object class provides five methods that are critical when writing multithreaded Java programs:

- notify
- notifyAll
- wait (three versions)

55.Define: Dynamic proxy.

A dynamic proxy is a class that implements a list of interfaces, which you specify at runtime when you create the proxy. To create a proxy, use the static method java.lang.reflect.Proxy::newProxyInstance().

This method takes three arguments:

- The class loader to define the proxy class
- An invocation handler to intercept and handle method calls
- A list of interfaces that the proxy instance implements

56.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.

57. Difference between Swing and Awt?

AWT are heavy-weight componenets. Swings are light-weight components. Hence swing works faster than AWT.

58. What is a layout manager and what are different types of layout managers available in java AWT?

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.

59. How are the elements of different layouts organized?

FlowLayout: The elements of a FlowLayout are organized in a top to bottom, left to right fashion.

BorderLayout: The elements of a BorderLayout are organized at the borders (North, South, East and West) and the center of a container.

CardLayout: The elements of a CardLayout are stacked, on top of the other, like a deck of cards. GridLayout: The elements of a GridLayout are of equal size and are laid out using the square of a grid.

GridBagLayout: The elements of a GridBagLayout are organized according to a grid. However, the elements are of different size and may occupy more than one row or column of the grid. In addition, the rows and columns may have different sizes. The default Layout Manager of Panel and Panel sub classes is FlowLayout.

60. What is skeleton and stub? what is the purpose of those?

Stub is a client side representation of the server, which takes care of communicating with the remote server. Skeleton is the server side representation. But that is no more in use… it is deprecated long before in JDK.

61. What is nested class?

If all the methods of a inner class is static then it is a nested class. 16. What is inner class? If the methods of the inner class can only be accessed via the instance of the inner class, then it is called inner class.

62. What is composition?

Holding the reference of the other class within some other class is known as composition. 20.What are the methods in Object?

clone, equals, wait, finalize, getClass, hashCode, notify, notifyAll, toString

63. What is source and listener?

source : A source is an object that generates an event. This occurs when the internal state of that object changes in some way.

listener : A listener is an object that is notified when an event occurs. It has two major requirements. First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications.

64.What is an exception?

An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

65.What is error?

An Error indicates that a non-recoverable condition has occurred that should not be caught. Error, a subclass of Throwable, is intended for drastic problems, such as OutOfMemoryError, which would be reported by the JVM itself.

66.Which is superclass of Exception?

"Throwable", the parent class of all exception related classes.

67.What are the advantages of using exception handling?

Exception handling provides the following advantages over "traditional" error management techniques:

- Separating Error Handling Code from "Regular" Code.
- Propagating Errors Up the Call Stack.
- Grouping Error Types and Error Differentiation.

68.What are the types of Exceptions in Java

There are two types of exceptions in Java, unchecked exceptions and checked exceptions.

Checked exceptions: A checked exception is some subclass of Exception (or Exception itself), excluding class RuntimeException and its subclasses. Each method must either handle all checked exceptions by supplying a catch clause or list each unhandled checked exception as a thrown exception.

Unchecked exceptions: All Exceptions that extend the RuntimeException class are unchecked exceptions. Class Error and its subclasses also are unchecked.

69. Why Errors are Not Checked?

A unchecked exception classes which are the error classes (Error and its subclasses) are exempted from compile-time checking because they can occur at many points in the program and recovery from them is difficult or impossible. A program declaring such exceptions would be pointlessly.

70. How does a try statement determine which catch clause should be used to handle an exception?

When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exception is executed. The remaining catch clauses are ignored.

71. What is the purpose of the finally clause of a try-catch-finally statement?

The finally clause is used to provide the capability to execute code no matter whether or not an exception is thrown or caught.

72. What is the difference between checked and Unchecked Exceptions in Java?

All predefined exceptions in Java are either a checked exception or an unchecked exception. Checked exceptions must be caught using try.. catch () block or we should throw the exception using throws clause. If you dont, compilation of program will fail.

73.What is the difference between exception and error?

The exception class defines mild error conditions that your program encounters. Exceptions can occur when trying to open the file, which does not exist, the network connection is disrupted, operands being manipulated are out of prescribed ranges, the class file you are interested in loading is missing. The error class defines serious error conditions that you should not attempt to recover from. In most cases it is advisable to let the program terminate when such an error is encountered.

74.What is the catch or declare rule for method declarations?

If a checked exception may be thrown within the body of a method, the method must either catch the exception or declare it in its throws clause.

75.When is the finally clause of a try-catch-finally statement executed?

The finally clause of the try-catch-finally statement is always executed unless the thread of execution terminates or an exception occurs within the execution of the finally clause.

76.What if there is a break or return statement in try block followed by finally block?

If there is a return statement in the try block, the finally block executes right after the return statement encountered, and before the return executes.

77.What are the different ways to handle exceptions?

There are two ways to handle exceptions:

- Wrapping the desired code in a try block followed by a catch block to catch the exceptions.

- List the desired exceptions in the throws clause of the method and let the caller of the method handle those exceptions.

78.What is the difference between throw and throws clause?

throw is used to throw an exception manually, where as throws is used in the case of checked exceptions, to tell the compiler that we haven't handled the exception, so that the exception will be handled by the calling function.

79.What are the different ways to generate and Exception?

There are two different ways to generate an Exception.

- Exceptions can be generated by the Java run-time system.
- Exceptions thrown by Java relate to fundamental errors that violate the rules of the Java language or the constraints of the Java execution environment.

80.Where does Exception stand in the Java tree hierarchy?

- java.lang.Object
- java.lang.Throwable
- java.lang.Exception
- java.lang.Error

81.Explain the exception hierarchy in java.

The hierarchy is as follows: Throwable is a parent class off all Exception classes. They are two types of Exceptions: Checked exceptions and UncheckedExceptions. Both type of exceptions extends Exception class

82.How do you get the descriptive information about the Exception occurred during the program execution?

All the exceptions inherit a method printStackTrace() from the Throwable class. This method prints the stack trace from where the exception occurred. It prints the most recently entered method first and continues down, printing the name of each method as it works its way down the call stack from the top.

83.Explain different way of using thread?

The thread could be implemented by using runnable interface or by inheriting from the Thread class. The former is more advantageous, 'cause when you are going for multiple inheritance..the only interface can help.

84.What are the different states of a thread ?

The different thread states are ready, running, waiting and dead.

85.Why are there separate wait and sleep methods?

The static Thread.sleep(long) method maintains control of thread execution but delays the next action until the sleep time expires.

The wait method gives up control over thread execution indefinitely so that other threads can run.

86.What is multithreading and what are the methods for inter-thread communication and what is the class in which these methods are defined?

Multithreading is the mechanism in which more than one thread run independent of each other within the process. wait (), notify () and notifyAll() methods can be used for inter-thread communication and these methods are in Object class. wait() : When a thread executes a call to wait() method, it surrenders the object lock and enters into a waiting state. notify() or notifyAll()

To remove a thread from the waiting state, some other thread must make a call to notify() or notifyAll() method on the same object.

87.What is synchronization and why is it important?

With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often leads to significant errors.

88.How does multithreading take place on a computer with a single CPU?

The operating system's task scheduler allocates execution time to multiple tasks. By quickly switching between executing tasks, it creates the impression that tasks execute sequentially.

89.What is the difference between process and thread?

Process is a program in execution whereas thread is a separate path of execution in a program.

8.What happens when you invoke a thread's interrupt method while it is sleeping or waiting?

When a task's interrupt() method is executed, the task enters the ready state. The next time the task enters the running state, an InterruptedException is thrown.

90.How can we create a thread?

A thread can be created by extending Thread class or by implementing Runnable interface. Then we need to override the method public void run().

91.What are three ways in which a thread can enter the waiting state?

A thread can enter the waiting state by invoking its sleep() method, by blocking on I/O, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait() method. It can also enter the waiting state by invoking its (deprecated) suspend() method.

92.How can i tell what state a thread is in ?

Prior to Java 5, isAlive() was commonly used to test a threads state. If isAlive() returned false the thread was either new or terminated but there was simply no way to differentiate between the two. 12.What is synchronized keyword? In what situations you will Use it?

Synchronization is the act of serializing access to critical sections of code. We will use this keyword when we expect multiple threads to access/modify the same data. To understand synchronization we need to look into thread execution manner.

93.What is serialization?

Serialization is the process of writing complete state of java object into output stream, that stream can be file or byte array or stream associated with TCP/IP socket.

94.What does the Serializable interface do?

Serializable is a tagging interface; it prescribes no methods. It serves to assign the Serializable data type to the tagged class and to identify the class as one which the developer has designed for persistence. ObjectOutputStream serializes only those objects which implement this interface.

95.When you will synchronize a piece of your code?

When you expect your code will be accessed by different threads and these threads may change a particular data causing data corruption.

96.What is daemon thread and which method is used to create the daemon thread?

Daemon thread is a low priority thread which runs intermittently in the back ground doing the garbage collection operation for the java runtime system. setDaemon method is used to create a daemon thread.

97.What is the difference between yielding and sleeping?

When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.

98.What is casting?

There are two types of casting, casting between primitive numeric types and casting between object references. Casting between numeric types is used to convert larger values, such as double values, to smaller values, such as byte values. Casting between object references is used to refer to an object by a compatible class, interface, or array type reference.

99.What classes of exceptions may be thrown by a throw statement?

A throw statement may throw any expression that may be assigned to the Throwable

type.

100.A Thread is runnable, how does that work?

The Thread class' run method normally invokes the run method of the Runnable type it is passed in its constructor. However, it is possible to override the thread's run method with your own.

101.What is thread priority?

Thread Priority is an integer value that identifies the relative order in which it should be executed with respect to others. The thread priority values ranging from 1- 10 and the default value is 5. But if a thread have higher priority doesn't means that it will execute first. The thread scheduling depends on the OS.

102.What are the different ways in which a thread can enter into waiting state?

There are three ways for a thread to enter into waiting state. By invoking its sleep() method, by blocking on I/O, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait() method.

103.How would you implement a thread pool?

The ThreadPool class is a generic implementation of a thread pool, which takes the following input Size of the pool to be constructed and name of the class which implements Runnable (which has a visible default constructor) and constructs a thread pool with active threads that are waiting for activation. once the threads have finished processing they come back and wait once again in the pool.

104.What is a thread group?

A thread group is a data structure that controls the state of collection of thread as a whole managed by the particular runtime environment.