MCQs

Q1Java Source Code is compiled into?

1. .Obj

2. Source Code

3. Byte Code

4. .Exe

correct: 3

reason: When we compile the Java program the compiler will convert source code to Byte Code nothing but .class files. Which is platform independent.

Q2Which of the following is not OOPS concept in Java?

1. Inheritance

2. Encapsulation

3. Polymorphism

4. Exception Handling

correct: 4

reason: There are 4 OOPS concepts in Java. Inheritance, Encapsulation, Polymorphism and Abstraction.

Q3Which statement is true about java?

1. Sequence dependent programming language

2. Platform independent programming language

3. Platform dependent programming language

4. Code dependent programming language

correct: 2

reason: Java is called ‘Platform Independent Language’ as it primarily works on the principle of ‘compile once, run everywhere’.

Q4Which component is responsible to optimize bytecode to machine code?

1. JVM

2. JDK

3. JIT

4. JRE

correct: 3

reason: JIT optimizes bytecode to machine specific language code by compiling similar bytecodes at the same time. This reduces overall time taken for compilation of bytecode to machine specific language.

Q5Which of the below is invalid identifier with the main method?

1. public

2. static

3. private

4. final

correct: 3

reason: main method cannot be private as it is invoked by external method. Other identifier are valid with main method.

Q6Which concept of Java is achieved by combining methods and attribute into a class?

1. Encapsulation

2. Inheritance

3. Polymorphism

4. Abstraction

correct: 1

reason: Encapsulation is implemented by combining methods and attribute into a class. The class acts like a container of encapsulating properties.

Q7Which of the following is a type of polymorphism in Java

1. Compile time polymorphism

2. Execution time polymorphism

3. Multiple polymorphism

4. Multilevel polymorphism

correct: 1

reason: There are two types of polymorphism in Java. Compile time polymorphism (overloading) and runtime polymorphism (overriding).

Q8In the beginning , Java was created in order to -

1. Create Strong Programming alternative to C++

2. Perform Operations on the Internet

3. Connect many household machines

4. Create high performance OS

correct: 3

reason: Basic aim of java was to solve the problem of connecting many household machines together.

Q9Which concept of Java is a way of converting real world objects in terms of class?

1. Polymorphism

2. Encapsulation

3. Abstraction

4. Inheritance

correct: 3

reason: Abstraction is the concept of defining real world objects in terms of classes or interfaces.

Q10What is it called if an object has its own lifecycle and there is no owner?

1. Aggregation

2. Composition

3. Encapsulation

4. Association

correct: 4

reason: It is a relationship where all objects have their own lifecycle and there is no owner. This occurs where many to many relationships are available, instead of one to one or one to many.

Q11What is it called where child object gets killed if parent object is killed?

1. Aggregation

2. Composition

3. Encapsulation

4. Association

correct: 2

reason: Composition occurs when child object gets killed if parent object gets killed. Composition is also known as strong Aggregation.

Q12The library containing files needed to integrate Java with programs written in other languages resides inside following sub folder.

1. bin

2. jre

3. include

4. lib

correct: 3

reason: The library containing files needed to integrate Java with programs written in other languages resides inside include sub folder.

Q13which of the following is not a Java keyword?

1. static

2. new

3. final

4. Integer

correct: 4

reason: Here static,new and final are keywords

Q14Decrement operator, –, decreases value of variable by what number?

1. 1

2. 2

3. 3

4. 4

correct: 1

reason: (-) operator value Decremented by 1

Q15Modulus operator, %, can be applied to which of these?

1. Integers

2. Floating – point numbers

3. Both Integers and floating – point numbers.

4. None of the mentioned

correct: 3

reason: Modulus operator can be applied to both integers and floating point numbers.

Q16Which of these values can a boolean variable contain?

1. True & False

2. 0 & 1

3. Any integer value

4. true

correct: 1

reason: Boolean variable can contain only one of two possible values, true and false.

Q17What is the range of data type byte in Java?

1. a). -128 to 127

2. b). -32768 to 32767

3. c). -2147483648 to 2147483647

4. d). None of the mentioned

correct: 1

reason: Byte occupies 8 bits in memory. Its range is from -128 to 127.

Q18What is the range of short data type in Java?

1. a) -128 to 127

2. b) -32768 to 32767

3. c) -2147483648 to 2147483647

4. d) None of the mentioned

correct: 2

reason: Short occupies 16 bits in memory. Its range is from -32768 to 32767.

Q19What is the output of this program?class booloperators {  
 public static void main(String args[])   
 {  
 boolean var1 = true;  
 boolean var2 = false;  
 System.out.println((var1 & var2));  
 }   
 }

1. a) 0

2. b) 1

3. c) true

4. d) false

correct: 4

reason: Explanation: boolean ‘&’ operator always returns true or false. var1 is defined true and var2 is defined false hence their ‘&’ operator result is false.

Q20Which one is a valid declaration of a boolean?

1. a) boolean b1 = 1;

2. b) boolean b2 = ‘false’;

3. c) boolean b3 = false;

4. d) boolean b4 = ‘true’

correct: 3

reason: Boolean can only be assigned true or false literals.

Q21What is the output of this program?  
  
 class asciicodes {  
 public static void main(String args[])   
 {  
 char var1 = 'A';  
 char var2 = 'a';  
 System.out.println((int)var1 + " " + (int)var2);  
 }   
 }

1. a) 162

2. b) 65 97

3. c) 67 95

4. d) 66 98

correct: 2

reason: ASCII code for ‘A’ is 65 and for ‘a’ is 97.

Q22What is the output of this program?  
  
 class mainclass {  
 public static void main(String args[])   
 {  
 char a = 'A';  
 a++;  
 System.out.print((int)a);  
 }   
 }

1. a) 66

2. b) 67

3. c) 65

4. d) 64

correct: 1

reason: ASCII value of ‘A’ is 65, on using ++ operator character value increments by one.

Q23Which one of these lists contains only Java programming language keywords?

1. a). class, if, void, long, Int, continue

2. b). goto, instanceof, native, finally, default, throws

3. c). try, virtual, throw, final, volatile, transient

4. d). strictfp, constant, super, implements, do

correct: 2

reason: All the words in option B are among the 49 Java keywords. Although goto reserved as a keyword in Java, goto is not used and has no function.  
  
Option A is wrong because the keyword for the primitive int starts with a lowercase i.  
  
Option C is wrong because "virtual" is a keyword in C++, but not Java.  
  
Option D is wrong because "constant" is not a keyword. Constants in Java are marked static and final.

Q24What will be the output of the program?  
  
class Equals   
{  
 public static void main(String [] args)   
 {  
 int x = 100;  
 double y = 100.1;  
 boolean b = (x = y); /\* Line 7 \*/  
 System.out.println(b);  
 }  
}

1. a). true

2. b). false

3. c). Compilation fails

4. d). An exception is thrown at runtime

correct: 3

reason: The code will not compile because in line 7, the line will work only if we use (x==y) in the line. The == operator compares values to produce a boolean, whereas the = operator assigns a value to variables.Option A, B, and D are incorrect because the code does not get as far as compiling. If we corrected this code, the output would be false.

Q25An expression involving byte, int and literal numbers is promoted to which of these?

1. int

2. long

3. byte

4. float

correct: 1

reason: No Explanation

Q26If an expression contains double, int, float, long, then whole expression will promoted into which of these data types?

1. long

2. int

3. double

4. float

correct: 3

reason: If any operand is double the result of expression is double.

Q27The conditional statement, ………………. can only test for equality, whereas ………………. can evaluate any type of Boolean expression.

1. if, switch

2. switch, if

3. while, if

4. if, while

correct: 2

reason: Switch statements checks for equality between the controlling variable and its constant cases.

Q28which of the following for loops will be an infinite loop?

1. for(;;)

2. for(i=0 ; i<1 ; i--)

3. for(i=0 ; ; i++)

4. all of the above

correct: 4

reason: No Explanation

Q29What is the output of this program?  
 class char\_increment {  
 public static void main(String args[])   
 {  
 char c1 = 'D';  
 char c2 = 84;  
 c2++;  
 c1++;  
 System.out.println(c1 + " " + c2);  
 }   
 }

1. E U

2. U E

3. V E

4. U F

correct: 1

reason: Operator ++ increments the value of character by 1. c1 and c2 are given values D and 84, when we use ++ operator their values increments by 1, c1 and c2 becomes E and U respectively.

Q30What is the error in this code?  
byte b = 50;  
 b = b \* 50;

1. b can not contain value 100, limited by its range.

2. \* operator has converted b \* 50 into int, which can not be converted to byte without casting.

3. b can not contain value 50.

4. No error in this code

correct: 2

reason: No Explanation

Q31int a=15;  
int b=25;  
if((a<b) || (a=5)>15)  
System.out.println(a);  
else  
System.out.println(b);

1. error

2. 15

3. 25

4. no output

correct: 2

reason: No Explanation

Q32What will be the output of following program?  
public class temp  
{  
 public static void main(String agrs[])  
 {  
 int i;  
 for(i=1; i<=10; i++);  
 System.out.print(i);  
 }  
}

1. 12345678910

2. 11

3. error

4. 1 2 3 4 5 6 7 8 9 10

correct: 2

reason: Consider the statement  
  
for(i=1; i<=10; i++);  
Since statement is terminated with semicolon (;), loop will be executed 10 times and then statement System.out.print(i); will be executed.

Q33What is the output of this program?  
  
class conversion {  
  
 public static void main(String args[])   
  
 {  
  
 double a = 295.04;  
  
 int b = 300;  
  
 byte c = (byte) a;  
  
 byte d = (byte) b;  
  
 System.out.println(c + " " + d);  
  
 }   
  
 }

1. 38 43

2. 39 44

3. 295 300

4. 295.04 300

correct: 2

reason: Type casting a larger variable into a smaller variable results in modulo of larger variable by range of smaller variable. b contains 300 which is larger than byte’s range i:e -128 to 127 hence d contains 300 modulo 256 i:e 44.

Q34What will be the output of the following program?  
class Test {   
public  
 static void main(String[] args)   
 {   
 int i = 0, j = 9;   
 do {   
 i++;   
 if (j-- < i++) {   
 break;   
 }   
 } while (i < 5);   
 System.out.println(i + "" + j);   
 }   
}

1. 44

2. 55

3. 66

4. 77

correct: 3

reason: we have to specially take care about the break statement. The execution of the program is going as usual as the control flow of do-while loop but whenever compiler encountered break statement its control comes out from the loop.

Q35What is the output of this program?  
class jump\_statments   
 {  
 public static void main(String args[])   
 {   
 int x = 2;  
 int y = 0;  
 for ( ; y < 10; ++y)   
 {  
 if (y % x == 0)   
 continue;   
 else if (y == 8)  
 break;  
 else  
 System.out.print(y + " ");  
 }  
 }   
 }

1. 1 3 5 7

2. 2 4 6 8

3. 1 3 5 7 9

4. 1 2 3 4 5 6 7 8 9

correct: 3

reason: Whenever y is divisible by x remainder body of loop is skipped by continue statement, therefore if condition y == 8 is never true as when y is 8, remainder body of loop is skipped by continue statements of first if. Control comes to print statement only in cases when y is odd.

Q36What is the output of this program?  
class comma\_operator   
 {  
 public static void main(String args[])   
 {   
 int sum = 0;  
 for (int i = 0, j = 0; i < 5 & j < 5; ++i, j = i + 1)  
 sum += i;  
 System.out.println(sum);  
 int var1 = 5;   
 int var2 = 6;  
 if ((var2 = 1) == var1)  
 System.out.print(var2);  
 else   
 System.out.print(++var2);  
 }   
 }

1. 5 1

2. 6 2

3. 14 3

4. compilation error 4

correct: 2

reason: Using comma operator, we can include more than one statement in the initialization and iteration portion of the for loop. Therefore both ++i and j = i + 1 is executed i gets the value – 0,1,2,3,4 & j gets the values -0,1,2,3,4,5.  
var2 is initialised to 1. The conditional statement returns false and the else part gets executed.

Q37What is false about constructor?

1. Constructors cannot be synchronized in Java

2. Java does not provide default copy constructor

3. Constructor can be overloaded

4.  “this” and “super” can be used in a constructor

correct: 3

reason: Default, parameterised constructors can be defined.

Q38Which of these is used as a default for a member of a class if no access specifier is used for it?

1. private

2. public

3. public, within its own package

4. protected

correct: 1

reason: When we pass an argument by call-by-value a copy of argument is made into the formal parameter of the subroutine and changes made on parameters of subroutine have no effect on original argument, they remain the same.

Q39Which of these classes are used by Byte streams for input and output operation

1. InputStream

2. InputOutputStream

3.  Reader

4. All the above

correct: 1

reason: Byte stream uses InputStream and OutputStream classes for input and output operation

Q40What is the process of defining two or more methods within same class that have same name but different parameters declaration?

1. method overloading

2. method overriding

3. method hiding

4. None

correct: 1

reason: Two or more methods can have same name as long as their parameters declaration is different, the methods are said to be overloaded and process is called method overloading. Method overloading is a way by which Java implements polymorphism.

Q41What is not the use of “this” keyword in Java

1. Passing itself to another method

2. Calling another constructor in constructor chaining

3. Referring to the instance variable when local variable has the same name

4. Passing itself to method of the same class

correct: 4

reason: “this” is an important keyword in java. It helps to distinguish between local variable and variables passed in the method as parameters.

Q42You want a class to have access to members of another class in the same package. Which is the most restrictive access that accomplishes this objective?

1. public

2. private

3. protected

4. default access

correct: 4

reason: To access the member of another class in same package in java default access is used.

Q43Which of these class contains the methods print() & println()?

1. System

2. Sytem.out

3. BufferedOutputStream

4. PrintStream

correct: 4

reason: print() and println() are defined under the class PrintStream, System.out is the byte stream used by these methods .

Q44How many arguments can be passed to main()?

1. Infinite

2. Only 1

3. System Dependent

4. None

correct: 1

reason: We can pass many arguments into a main function as many no of times

Q45which object will be created first? class student{ int marks; }; student s1, s2,s3;

1. s1 then s2 then s3

2. s3 then s2 then s1

3. s2 then s3 then s1

4. All are created at same time

correct: 1

reason: Explicit call to a constructor can let you create temporary instance. This is because the temporary instances doesn’t have any name. Those are deleted from memory as soon as their reference is removed.

Q46Which of the following class level (nonlocal) variable declarations will not compile?

1. protected int a;

2. transient int b=3;

3. public static final int c;

4. private synchronized e;

correct: 4

reason: non local variable in the class declaration defined by private will not be compiled

Q47what is the output of this program? Class Output{ Public static void main(String args[]) { StringBuffer s1 = new StringBuffer("Hello");s1.setCharAt(1,x);System.out.println(s1);}}

1.  xello

2.  xxxxx

3. Hxllo

4. Hexlo

correct: 3

reason: $ javac Output.java $ java Output Hxllo

Q48What is the output of this program, Command line execution is done as – “java Output command Line 10 A b 4 N”?class Output {  
 public static void main(String args[]) {  
 System.out.print("(int)args[2] \* 2");  
 }  
 }

1. Java

2. 10

3. 20

4. b

correct: 3

reason: $ javac Output.java $ java Output 20

Q49When Overloading does not occur?

1. More than one method with same name but different method signature and different number or type of parameters

2. More than one method with same name, same signature but different number of signature

3. More than one method with same name, same signature, same number of parameters but different type

4. More than one method with same name, same number of parameters and type but different signature

correct: 4

reason: Overloading occurs when more than one method with same name but different constructor and also when same signature but different number of parameters and/or parameter type.

Q50What allows the programmer to destroy an object x?

1. x.delete()

2. x.finalize()

3. Runtime.getRuntime().gc()

4. Only the garbage collection system can destroy an object.

correct: 4

reason: No Explanation

Q51What would be the behaviour if this() and super() used in a method?

1. Runtime error

2. Throws exception

3. compile time error

4. Runs successfully

correct: 3

reason: this() and super() cannot be used in a method. This throws compile time error.

Q52Which of these constructors is used to create an empty String object?

1. String()

2. String(void)

3. String(0)

4. None of the mentioned

correct: 1

reason: String()

Q53void start() {   
 A a = new A();   
 B b = new B();   
 a.s(b);   
 b = null; /\* Line 5 \*/  
 a = null; /\* Line 6 \*/  
 System.out.println("start completed"); /\* Line 7 \*/  
}   
When is the B object, created in line 3, eligible for garbage collection?

1. after line 5

2. after line 6

3. after line 7

4. There is no way to be absolutely certain.

correct: 4

reason: No Explanation

Q54Which of these is not a correct statement?

1. A recursive method must have a base case

2. Recursion always uses stack

3. Recursive methods are faster that programmers written loop to call the function repeatedly using a stack

4. Recursion is managed by Java Runtime environment

correct: 4

reason: Recursion is always managed by operating system

Q55Abstract class cannot have a constructor.

1. TRUE

2. FALSE

3. 1

4. 0

correct: 2

reason: No instance can be created of abstract class. Only pointer can hold instance of object.

Q56What is the output of this program?  
class String\_demo   
 {  
 public static void main(String args[])  
 {  
 int ascii[] = { 65, 66, 67, 68};  
 String s = new String(ascii, 1, 3);  
 System.out.println(s);  
 }  
 }

1. ABC

2. BCD

3. CDA

4. ABCD

correct: 2

reason: ascii is an array of integers which contains ascii codes of Characters A, B, C, D. String(ascii, 1, 3) is an constructor which initializes s with Characters corresponding to ascii codes stored in array ascii, starting position being given by 1 & ending position by 3, Thus s stores BCD.

Q57public class X   
{  
 public static void main(String [] args)   
 {  
 X x = new X();  
 X x2 = m1(x); /\* Line 6 \*/  
 X x4 = new X();  
 x2 = x4; /\* Line 8 \*/  
 doComplexStuff();  
 }  
 static X m1(X mx)   
 {  
 mx = new X();  
 return mx;  
 }  
}  
After line 8 runs. how many objects are eligible for garbage collection?

1. 0

2. 1

3. 2

4. 3

correct: 2

reason: No Explanation

Q58What is the output of this program?  
class recursion   
 {  
 int func (int n)   
 {  
 int result;  
 if (n == 1)  
 return 1;  
 result = func (n - 1);  
 return result;  
 }  
 }   
 class Output   
 {  
 public static void main(String args[])   
 {  
 recursion obj = new recursion() ;  
 System.out.print(obj.func(5));  
 }  
 }

1. 0

2. 1

3. 120

4.  None of the mentioned

correct: 2

reason: No Explanation

Q59public class Test  
{  
public static void main(String args[])  
{  
String a="sai";  
String b="ram"  
System.out.println(a.charAt(0) > b.charAt(0) );  
}  
}  
Determine Output

1. TRUE

2. FALSE

3. Compilation error

4. throws Exception

correct: 1

reason: Output will be True. Since when s1.charAt(0) ascii value is greater then s2.charAt(0). So it will return True.

Q60What is the output of this program?  
  
 class String\_demo   
{  
 public static void main(String args[])  
 {  
 char chars[] = {'a', 'b', 'c'};  
 String s = new String(chars);  
 String s1 = "abcd";  
 int len1 = s1.length();  
 int len2 = s.length();  
 System.out.println(len1 + " " + len2);  
 }  
 }

1. 3 0

2. 0 3

3. 3 4

4. 4 3

correct: 4

reason: No Explanation

Q61In inheritance, the class from which the properties are inherited is known as ?

1. Upper class

2. child class

3. Parent class

4. Top class

correct: 3

reason: from the parent class properties are inherited to child class

Q62In inheritance, the class which inherits the properties of another class is known as ?

1. lower class

2. child class

3. down class

4. None

correct: 2

reason: the properties come down from parent to child class

Q63which keyword is used for inheritance in java ?

1. super

2. this

3. implements

4. extends

correct: 4

reason: we use extends keyword for inheritance

Q64inheritance gives scope for \_\_\_\_\_\_

1. reusability

2. GUI apps

3. arrays

4. None

correct: 1

reason: reusing of existing modules is possible through inheritance

Q65What kind of inheritance is not possible in java ?

1. multiple

2. single

3. double

4. triple

correct: 1

reason: java doesn't support multiple inheritance directly.

Q66Using which of the following, multiple inheritance in Java can be implemented?

1. Protected methods

2. Multithreading

3. interfaces

4. Private methods

correct: 3

reason: interfaces solve the problem of multiple inheritance in java

Q67All classes in Java are inherited from which class?

1. java.lang.class

2.  java.class.inherited

3. java.class.object

4. java.lang.Object

correct: 4

reason: All classes in java are inherited from Object class. Interfaces are not inherited from Object Class.

Q68In order to restrict a variable of a class from inheriting to subclass, how variable should be declared?

1. Protected

2. Private

3. Public

4. Static

correct: 2

reason: By declaring variable private, the variable will not be available in inherited to subclass.

Q69If super class and subclass have same variable name, which keyword should be used to use super class variable ?

1. super

2. this

3. upper

4. classname

correct: 1

reason: Super keyword is used to access hidden super class variable in subclass.

Q70Static members are not inherited to subclass.

1. TRUE

2. both

3. none

4. FALSE

correct: 4

reason: Static members are also inherited to subclasses.

Q71Which of the following is used for implementing inheritance through an interface?

1. inherited

2. using

3. extends

4. implements

correct: 4

reason: Static members are also inherited to subclasses.

Q72Which of the following is used for implementing inheritance through class?

1. inherited

2. using

3. extends

4. implements

correct: 3

reason: Class can be extended using extends keyword. One class can extend only one class. A final class cannot be extended.

Q73 What is the output of the following java program?  
public class Outer   
{   
 public static int temp1 = 1;   
 private static int temp2 = 2;   
 public int temp3 = 3;   
 private int temp4 = 4;   
   
 public static class Inner   
 {   
 private static int temp5 = 5;   
   
 private static int getSum()   
 {   
 return (temp1 + temp2 + temp3 + temp4 + temp5);   
 }   
 }   
   
 public static void main(String[] args)   
 {   
 Outer.Inner obj = new Outer.Inner();   
 System.out.println(obj.getSum());   
 }   
   
}

1. 15

2. 9

3. 5

4. Compilation Error

correct: 4

reason: static inner classes cannot access non-static fields of the outer class.

Q74public class Outer   
{   
 private static int data = 10;   
 private static int LocalClass()   
 {   
 class Inner   
 {   
 public int data = 20;   
 private int getData()   
 {   
 return data;   
 }   
 };   
 Inner inner = new Inner();   
 return inner.getData();   
 }   
   
 public static void main(String[] args)   
 {   
 System.out.println(data \* LocalClass());   
 }   
}

1. Compilation error

2. Runtime Error

3. 200

4. None of the above

correct: 3

reason: LocalClass() method defines a local inner class. This method creates an object of class Inner and return the value of the variable data that resides within it.

Q75interface Anonymous   
{   
 public int getValue();   
}   
public class Outer   
{   
 private int data = 15;   
 public static void main(String[] args)   
 {   
 Anonymous inner = new Anonymous()   
 {   
 int data = 5;   
 public int getValue()   
 {   
 return data;   
 }   
 public int getData()   
 {   
 return data;   
 }   
 };   
 Outer outer = new Outer();   
 System.out.println(inner.getValue() + inner.getData() + outer.data);   
 }   
}

1. 25

2. Compilation error

3. 20

4. Runtime error

correct: 2

reason: the method getData() is undefined in Anonymous class which causes the compilation error.

Q76  
public class Outer   
{   
 private int data = 10;   
   
 class Inner   
 {   
 private int data = 20;   
 private int getData()   
 {   
 return data;   
 }   
 public void main(String[] args)   
 {   
 Inner inner = new Inner();   
 System.out.println(inner.getData());   
   
 }   
 }   
 private int getData()   
 {   
 return data;   
 }   
 public static void main(String[] args)   
 {   
 Outer outer = new Outer();   
 Outer.Inner inner = outer.new Inner();   
 System.out.printf("%d", outer.getData());   
 inner.main(args);   
 }   
}

1. 2010

2. 1020

3. Compilation Error

4. None of these

correct: 3

reason: Inner class defined above though, have access to the private variable data of the Outer class, but declaring a variable data inside an inner class makes it specific to the Inner class with no conflicts in term of variable declaration.

Q77The nested class can be declared \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Public

2. Private

3. Protected

4. Public, Protected, Private or default

correct: 4

reason: The nested class can be declared with any specifier, unlike the outer classes which can only be declared public or package private. This is a flexibility given for the nested class being a member of enclosing class.

Q78 How to create object of the inner class?

1. OuterClass.InnerClass innerObject = new OuterClass.InnerClass ();

2. OuterClass.InnerClass innerObject = new InnerClass();

3. InnerClass innerObject = outerObject.new InnerClass();

4. OuterClass.InnerClass = outerObject.new InnerClass();

correct: 1

reason: An instance of inner class can exist only within instance of outer class. To instantiate the inner class, one must instantiate the outer class first. This can be done by the correct syntax above.

Q79How to access static nested classes?

1. OuterClass.StaticNestedClass

2. OuterClass->StaticNestedClass

3. OuterClass(StaticNestedClass)

4. OuterClass[StaticNestedClass].

correct: 1

reason: Like any other member of the class, the static nested class uses the dot operator to be accessed. The reason behind is, the static classes can’t work with instances, hence we use enclosing class name to access static nested class.

Q80Instance of inner class can exist only \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ enclosing class.

1. Within

2. Outside

3. Private to

4. Public to

correct: 1

reason: The class defined inside another class is local to the enclosing class. This means that the instance of inner class will not be valid outside the enclosing class. There is no restriction for instance to be private or public always.

Q81Which feature of OOP reduces the use of nested classes?

1. Encapsulation

2. Inheritance

3. Binding

4. Abstraction

correct: 2

reason: Using inheritance we can have the security of the class being inherited. The subclass can access the members of parent class. And have more feature than a nested class being used.

Q82Use of nested class \_\_\_\_\_\_\_\_\_\_\_\_ encapsulation.

1. Increases

2. Decreases

3. Doesn’t affect

4. Slightly decreases

correct: 1

reason: The use of nested class increases encapsulation as the inner class is getting even more grouped into the enclosing class. Firstly the class encapsulate the data, having nested classes can increase the encapsulation even further.

Q83Which among the following is correct advantage of nested classes?

1. Makes the code more complex

2. Makes the code unreadable

3. Makes the code efficient and readable

4. Makes the code multithreaded

correct: 3

reason: The use of nested classes make the code more streamed towards a single concept. This allows to group the most similar and related classes together and makes it even more efficient and readable.

Q84If a declaration of a member in inner class has the same name as that in the outer class, then \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ enclosing scope.

1. Outer declaration shadows inner declaration in

2. Inner declaration shadows outer declaration in

3. Declaration gives compile time error

4. Declaration gives runtime error

correct: 2

reason: The inner class will have more preference for its local members than those of the enclosing members. Hence it will shadow the enclosing class members. This process is known as shadowing.

Q85Which of these keywords is used by a class to use an interface defined previously?

1. imports

2. Import

3. implements

4. Implement

correct: 3

reason: implements is the keyword to implement a class

Q86Which two of the following are legal declarations for abstract classes and interfaces? 1. final abstract class Test {} 2. public static interface Test {} 3. final public class Test {} 4. protected abstract class Test {} 5. protected interface Test {} 6. abstract public class Test {}  
1. final abstract class Test {}

1. 1 and 2

2. 2 and 4

3. 3 and 5

4. 3 and 6

correct: 4

reason: both are public. Any one can provide implementation to them

Q87Which of the following is FALSE about abstract classes in Java

1. If we derive an abstract class and do not implement all the abstract methods, then the derived class should also be marked as abstract using 'abstract' keyword

2. Abstract classes can have constructors

3. A class can be made abstract without any abstract method

4. A class can inherit from multiple abstract classes

correct: 4

reason: multiple inheritence is not supported w.r.t classes

Q88 What is an interface?

1. An interface is a collection of constants and method declarations

2. An interface is a class that a child class can extend.

3. An interface is a collection of GUI components.

4. An interface is the collection of public methods of a class.

correct: 1

reason: interface definition

Q89Which of the following is true about interfaces in java.   
1) An interface can contain following type of members.  
....public, static, final fields (i.e., constants)  
2) An instance of interface can be created.  
3) A class can implement multiple interfaces.  
4) Many classes can implement the same interface.

1. 1, 3 and 4

2. 1, 2 and 4

3. 2, 3 and 4

4. 1, 2, 3 and 4

correct: 1

reason: interface can not be intantiated

Q90Which of the following statement(s) with regard to an abstract class in JAVA is/are TRUE ? I. An abstract class is one that is not used to create objects. II. An abstract class is designed only to act as a base class to be inherited by other classes.

1. Only I

2. Only II

3. Neither I nor II

4. Both I and II

correct: 4

reason: Abstract data class is not used to create objects in Java and it is designed only to act as a base class to be inherited by other classes. Both Statement are correct. For more information Refer:Abstract Classes in Java Option (D) is correct

Q91interface Taxable { double taxRate = 0.06; double calculateTax(); }

1. No---because it contains a variable and interfaces cannot contain variables.

2. No---because the interface cannot contain a method that returns a value

3.  Yes--taxRate will automatically be a constant since it is in an interface.

4. Yes--the method body will automatically be filled in.

correct: 3

reason: all variables in interface are constant by default

Q92Which of the following is true?

1. A child class can extend a parent or implement an interface, but not do both

2. A child class can extend just one parent and can implement just one interface.

3. A child class can extend just one parent and can implement zero or more interfaces.

4. A child class can extend zero or more parents, and can implement zero or more interfaces.

correct: 3

reason: multiple inheritence is not supported w.r.t classes and supported w.r.t interfaces

Q93Can an interface name be used as the type of a variable, like this:   
public static void main( String[] args ) { SomeInterface x; ... }

1. No---a variable must always be an object reference type.

2. No---a variable must always be an object reference type or a primitive type.

3.  No---a variable must always be a primitive type.

4. Yes--the variable can refer to any object who's class implements the interface.

correct: 4

reason: feature of interface

Q94Which of the following is correct way of implementing an interface salary by class manager?

1.  class manager extends salary {}

2. class manager implements salary {}

3. class manager imports salary {}

4. None of the mentioned.

correct: 2

reason: class manager implements salary {}

Q95Can an interface extend another interface?

1. No---only classes can be extended.

2. No---interfaces can not be part of a hierarchy.

3. Yes--since all interfaces automatically extend Object.

4. Yes

correct: 4

reason: inheritence w.r.t interface

Q96Which if the following is correct

1. abstract class can not have non abstract methods

2. abstract class should have abstract methods

3. abstract class can have both abstract and non abstract methods

4. abstract class can be instantiated

correct: 3

reason: feature of abstract class

Q97Given the following piece of code:   
public class School{ public abstract double numberOfStudent(); }

1. The keywords public and abstract cannot be used together.

2. The method numberOfStudent() in class School must have a body

3. You must add a return statement in method numberOfStudent().

4. Class School must be defined abstract.

correct: 4

reason: abstract class definition

Q98Which of the following class definitions defines a legal abstract class?

1. class A { abstract void unfinished() { } }

2. class A { abstract void unfinished(); }

3. abstract class A { abstract void unfinished(); }

4. public class abstract A { abstract void unfinished(); }

correct: 3

reason: abstract keyword should be before class

Q99Suppose A is an abstract class, B is a concrete subclass of A, and both A and B have a default constructor. Which of the following is correct? 1. A a = new A(); 2. A a = new B(); 3. B b = new A(); 4. B b = new B();

1. 1 and 2

2. 2 and 4

3. 3 and 4

4. 1 and 3

correct: 2

reason: parent class reference can hold all of its child classes objects

Q100Which of the following is/are true about packages in Java? 1 ) Every class is part of some package. 2) All classes in a file are part of the same package. 3) If no package is specified, the classes in the file go into a special unnamed package 4) If no package is specified, a new package is created with folder name of class and the class is put in this package.

1. Both Type 1 and Type 2

2. Both Type 1 and Type 3

3. Both Type 3 and Type 4

4. Type 4 only

correct: 1

reason: Package in java is designed for providing a way to keep one set of names separate from another. The class names declared in one Package does not conflict with the same class names declared in another

Q101Which of these is a mechanism for naming and visibility control of a class and its content?

1. Object

2. Packages

3. Interfaces

4. None of the Mentioned.

correct: 2

reason: Packages are both naming and visibility control mechanism. We can define a class inside a package which is not accessible by code outside the package.

Q102Exception class is in \_\_\_\_\_\_\_ package

1. java.file

2. java.io

3. java.lang

4. java.util

correct: 3

reason: No Explanation

Q103Which of this package is used for invoking a method remotely?

1. java.rmi

2. java.awt

3. java.util

4. java.util

correct: 1

reason: java.rmi provides capabilities for remote method invocation.

Q104Which of this package is used for analyzing code during run-time?

1. java.applet

2. java.awt

3. java.io

4. java.lang.reflect

correct: 4

reason: Reflection is the ability of a software to analyze itself. This is provided by java.lang.reflect package.

Q105A method within a class is only accessible by classes that are defined within the same package as the class of the method. Which one of the following is used to enforce such restriction?

1. Declare the method with the keyword public

2. Declare the method with the keyword private

3. Declare the method with the keyword protected

4. Do not declare the method with any accessibility modifiers

correct: 4

reason: No Explanation

Q106File class is included in which package?

1. java.io package

2. java.lang package

3. java.awt package

4. java.net.package

correct: 1

reason: No Explanation

Q107Which package contain the JDBC classes

1. java.jdbc.

2. java.sql.jdbc

3. java.sql

4. javax.jdbc.sql

correct: 3

reason: No Explanation

Q108What would be the output if the following code is executed?  
  
import java.io.\*;  
protected class MyClass  
{  
Public static void main(String args[]){  
System.out.println(“My First Class”);  
}  
}

1. The code will not compile because the class cannot be 'protected';

2. code will not compile successfully but will not print anything on standard output

3.    The program will compile and be executed successfully, print the message My First Clas.

4.   The code will not be compile because the main method cannot be written in a protected class

correct: 1

reason: No Explanation

Q109What is the process of defining two or more methods within same class that have same name but different parameters declaration?

1. method overloading

2. method overriding

3.  method hiding

4. none of the mentioned

correct: 1

reason: Two or more methods can have same name as long as their parameters declaration is different, the methods are said to be overloaded and process is called method overloading. Method overloading is a way by which Java implements polymorphism.

Q110What is the process of defining a method in a subclass having same name & type signature as a method in its superclass?

1. Method overloading

2. Method overriding

3. Method hiding

4. None of the mentioned

correct: 2

reason: No Explanation

Q111 Which of these keywords can be used to prevent Method overriding?

1. static

2. constant

3. protected

4.  final

correct: 4

reason: To disallow a method from being overridden, specify final as a modifier at the start of its declaration. Methods declared as final cannot be overridden.

Q112 Which of these is supported by method overriding in Java?

1. Abstraction

2. Encapsulation

3. Polymorphism

4. None of the mentioned

correct: 3

reason: No Explanation

Q113What concepts come under Polymorphism in java?

1. Method overloading

2. Constructor overloading

3. . Method overriding

4. All the above

correct: 4

reason: No Explanation

Q114What is an advantage of polymorphism?

1.   The same program logic can be used with objects of several related types.

2. Variables can be re-used in order to save memory.

3.   Constructing new objects from old objects of a similar type saves time.

4. .    Polymorphism is a dangerous aspect of inheritance and should be avoided.

correct: 1

reason: No Explanation

Q115Can an object of a child type be assigned to a variable of the parent type? For example,   
Card crd;  
BirthDay bd = new BirthDay("Lucinda", 42);  
  
crd = bd; // is this correct?

1. No--there must always be an exact match between the variable and the object types.

2. No--but a object of parent type can be assigned to a variable of child type

3. Yes--an object can be assigned to a reference variable of the parent type

4. Yes--any object can be assgned to any reference variable.

correct: 3

reason: No Explanation

Q116Can an abstract class define both abstract methods and non-abstract methods?

1.   No--it must have all one or the other.

2. No--it must have all abstract methods.

3.     Yes--but the child classes do not inherit the abstract methods.

4.   Yes--the child classes inherit both.

correct: 4

reason: No Explanation

Q117What is an abstract method?

1. An abstract method is any method in an abstract class.

2. .    An abstract method is a method which cannot be inherited.

3. An abstract method is one without a body that is declared with the reserved word abstract.

4. .    An abstract method is a method in the child class that overrids a parent method.

correct: 3

reason: No Explanation

Q118What will be the output of the following program code?  
class Rectangle{  
 public int area(int length, int width){  
 return length\*width;  
 }  
}  
class Square extends Rectangle{  
 public int area(long length, long width){  
 return (int) Math.pow(length, 2);  
 }  
}  
public class Test{  
 public static void main(String args[]){  
 Square r = new Square();  
 System.out.println(r.area(5 , 4));  
 }  
}

1. Will not compile.

2. Will compile and run printing out 20

3. Runtime error

4. Will compile and run printing out 25

correct: 2

reason: No Explanation

Q119What will be the output?  
class A{  
 static void method(){  
 System.out.println("Class A method");  
 }  
}  
class B extends A{  
 static void method(){  
 System.out.println("Class B method");  
 }  
}  
public class Test{  
 public static void main(String args[]){  
 A a = new B();  
 a.method();  
 }  
}

1. Class A method

2.  Class B method

3.  Compilation Error

4.  Runtime Error

correct: 1

reason: Overriding in Java simply means that the particular method would be called based on the run time type of the object and not on the compile time type. But in the above case the methods are static which means access to them is always resolved during compile time only using the compile time type information. Accessing them using object references is just an extra liberty given by the designers of Java.

Q120class GFG   
{   
 protected void getData()   
 {   
 System.out.println("Inside GFG");   
 }   
}   
class GeeksforGeeks extends GFG   
{   
 protected void getData()   
 {   
 System.out.println("Inside GeeksforGeeks");   
 }   
}   
  
public class Test   
{   
 public static void main(String[] args)   
 {   
 GFG obj = new GeeksforGeeks();   
 obj.getData();   
 }   
}

1.  Inside GFG

2. Inside GeeksforGeeks

3. Compilation error

4. Runtime error

correct: 2

reason: A reference variable of GFG class is used to point to an object of class GeeksforGeeks. At the time of compilation, the JVM checks whether the method being called is defined in GFG class, but at the runtime, JVM invoke the method of GeeksforGeeks class because the object is from class GeeksforGeeks. Refer to static vs dynamic binding in java for more details.

Q121class Test   
{   
 void myMethod()   
 {   
 System.out.println("GeeksforGeeks");   
 }   
}   
public class Derived extends Test   
{   
 void myMethod()   
 {   
 System.out.println("GFG");   
 }   
   
 public static void main(String[] args)   
 {   
 Derived object = new Test();   
 object.myMethod();   
 }   
}

1. GeeksforGeeks

2. GFG

3. Compilation error

4. Runtime error

correct: 3

reason: A child class cannot be used as a reference to an object of super class.

Q122When does Exceptions in Java arises in code sequence?

1. Run Time

2. Compilation Time

3. Can Occur Any Time

4. None of the mentioned

correct: 1

reason: Exceptions in java are run-time errors.

Q123Which of these keywords is not a part of exception handling?

1. try

2. finally

3. thrown

4. catch

correct: 3

reason: Exceptional handling is managed via 5 keywords – try, catch, throws, throw and finally.

Q124Which of these keywords must be used to monitor for exceptions?

1. try

2. finally

3. throw

4. catch

correct: 1

reason: No Explanation

Q125Which of these keywords must be used to handle the exception thrown by try block in some rational manner?

1. try

2. finally

3. throw

4. catch

correct: 4

reason: If an exception occurs within the try block, it is thrown and cached by catch block for processing.

Q126Which of these keywords is used to manually throw an exception?

1. try

2. finally

3. throw

4. catch

correct: 3

reason: No Explanation

Q127Which of these is a super class of all exceptional type classes?

1. String

2. RuntimeExceptions

3. Throwable

4. Cachable

correct: 3

reason: All the exception types are subclasses of the built in class Throwable.

Q128Which of these class is related to all the exceptions that can be caught by using catch?

1. Error

2. Exception

3. RuntimeExecption

4. All of the mentioned

correct: 2

reason: Error class is related to java run time error that can’t be caught usually, RuntimeExecption is subclass of Exception class which contains all the exceptions that can be caught.

Q129Which of these class is related to all the exceptions that cannot be caught?

1. Error

2. Exception

3. RuntimeExecption

4. All of the mentioned

correct: 1

reason: Error class is related to java run time error that can’t be caught usually, RuntimeExecption is subclass of Exception class which contains all the exceptions that can be caught.

Q130Which of these handles the exception when no catch is used?

1. Default handler

2. finally

3. throw handler

4. Java run time system

correct: 1

reason: No Explanation

Q131Which of the following operators is used to generate instance of an exception which can be thrown using throw?

1. thrown

2. alloc

3. malloc

4. new

correct: 4

reason: new operator is used to create instance of an exception. Exceptions may have parameter as a String or have no parameter.

Q132Which of these keywords is used to by the calling function to guard against the exception that is thrown by called function?

1. try

2. throws

3. throw

4. catch

correct: 2

reason: If a method is capable of causing an exception that it does not handle. It must specify this behaviour the behaviour so that callers of the method can guard themselves against that exception. This is done by using throws clause in methods declaration.

Q133At runtime, error is recoverable.

1. TRUE

2. FALSE

3. nan

4. nan

correct: 2

reason: Error is not recoverable at runtime. The control is lost from the application.

Q134A single try block must be followed by which of these?

1. finally

2. catch

3. finally & catch

4. none of the mentioned

correct: 3

reason: try block can be followed by any of finally or catch block, try block checks for exceptions and work is performed by finally and catch block as per the exception.

Q135Which of these clause will be executed even if no exceptions are found?

1. throws

2. finally

3. throw

4. catch

correct: 2

reason: finally keyword is used to define a set of instructions that will be executed irrespective of the exception found or not.

Q136What will be the output of the program  
public class Foo{  
public static void main(String args[]){  
try{  
return;  
}finally{  
System.out.println(“Finally”);  
}  
}  
}

1. finally

2. Compilation fails

3. code runs with no output

4. an exception is thrown at runtime

correct: 1

reason: If you put a finally block after a try and its associated catch blocks, then once execution enters the try block, the code in that finally block will definitely be executed except in the following circumstances:   
1. An exception arising in the finally block itself.   
2. The death of the thread.   
3. The use of System.exit()   
4. Turning off the power to the CPU.

Q137What will be the output of the program  
try   
{   
 int x = 0;   
 int y = 5 / x;   
}   
catch (Exception e)   
{  
 System.out.println("Exception");   
}   
catch (ArithmeticException ae)   
{  
 System.out.println(" Arithmetic Exception");   
}   
System.out.println("finished");

1. finished

2. Exception

3. compilation fails

4. Arithmetic Exceptional

correct: 3

reason: Compilation fails because ArithmeticException has already been caught. ArithmeticException is a subclass of java.lang.Exception, by time the ArithmeticException has been specified it has already been caught by the Exception class.  
If ArithmeticException appears before Exception, then the file will compile. When catching exceptions the more specific exceptions must be listed before the more general (the subclasses must be caught before the superclasses).

Q138Exception class is in ----------------package

1. java.io

2. java.file

3. java.lang

4. java.util

correct: 3

reason: java.lang.Throwable  
public class Exception extends Throwable   
The class Exception and its subclasses are a form of Throwable that indicates conditions that a reasonable application might want to catch.

Q139To create our own exception class, we have to

1. Extend exception class

2. create our own try and catch block

3. use finally block

4. use throws keywords

correct: 1

reason: Java provides us facility to create our own exceptions which are basically derived classes of Exception.  
Example  
class MyException extends Exception{}

Q140 Which of the following must be true of the object thrown by a throw statement?

1. It must be assignable to the throwable type

2. It must be assignable to the error type

3. It must be assignable to the Exception type

4. It must be assignable to the string type

correct: 1

reason: The throw statement requires a single argument: a throwable object. Throwable objects are instances of any subclass of the Throwable class. Here’s an example of a throw statement:   
throw someThrowableObject;

Q141FileNotFoundException

1. is a subclass/extends IOException

2. is a compile time exception

3. Found in java.io package

4. All

correct: 4

reason: public class FileNotFoundException extends IOException  
Signals that an attempt to open the file denoted by a specified pathname has failed.   
This exception will be thrown by the FileInputStream, FileOutputStream, and RandomAccessFile constructors when a file with the specified pathname does not exist. It will also be thrown by these constructors if the file does exist but for some reason is inaccessible, for example when an attempt is made to open a read-only file for writing.

Q142Whic of these are java.lang.Error in exception handling in java?

1. VirtualMachineError

2. ALL

3. AssertionError

4. ThreadDeath

correct: 2

reason: An Error is a subclass of Throwable that indicates serious problems that a reasonable application should not try to catch. Most such errors are abnormal conditions. The ThreadDeath error, though a "normal" condition, is also a subclass of Error because most applications should not try to catch it.

Q143Which is valid about java.lang.Exceptions?

1. The class Exception and all its subclasses that are not also subclasses of RuntimeException are checked exceptions

2. The class error and all its subclasses are unchecked exceptions

3. The class RuntimeException and all its subclasses are unchecked exceptions

4. All

correct: 4

reason: public class Exception extends Throwable  
The class Exception and its subclasses are a form of Throwable that indicates conditions that a reasonable application might want to catch.   
The class Exception and any subclasses that are not also subclasses of RuntimeException are checked exceptions. Checked exceptions need to be declared in a method or constructor's throws clause if they can be thrown by the execution of the method or constructor and propagate outside the method or constructor boundary.

Q144can a method be overloaded on the basis of exception?

1. Will produce runtime Error

2. will produce compiletime error – ambiguity error

3. yes can be overloaded on basis of exception

4. None

correct: 3

reason: The overloaded method can throw any exception thrown / not thrown by the method it overloads because both are independent things.

Q145What is the output of this program?  
Class exception\_handling   
 {  
 public static void main(String args[])  
 {  
 try   
 {  
 int a = 1;  
 int b = 10 / a;  
 try   
 {  
 if (a == 1)  
 a = a / a - a;  
 if (a == 2)   
 {  
 int c[] = {1};  
 c[8] = 9;  
 }  
 }  
 finally   
 {  
 System.out.print("A");  
 }  
 }  
 catch (Exception e)   
 {  
 System.out.println("B");  
 }  
 }  
 }

1. A

2. B

3. AB

4. BA

correct: 1

reason: The inner try block does not have a catch which can tackle ArrayIndexOutOfBoundException hence finally is executed which prints ‘A’ the outer try block does have catch for ArrayIndexOutOfBoundException exception but no such exception occurs in it hence its catch is never executed and only ‘A’ is printed.

Q146What is the output of this program?  
class exception\_handling   
 {  
 public static void main(String args[])   
 {  
 try   
 {  
 throw new NullPointerException ("Hello");  
 }  
 catch(ArithmeticException e)  
 {  
 System.out.print("B");   
 }  
 }  
 }

1. A

2. B

3. Compilation Error

4. RUNTIME ERROR

correct: 4

reason: Try block is throwing NullPointerException but the catch block is used to counter Arithmetic Exception. Hence NullPointerException occurs since no catch is there which can handle it, runtime error occurs.

Q147Which of these exceptions will occur if we try to access the index of an array beyond its length?

1. ArithmeticException

2. ArrayException

3. ArrayIndexException

4. ArrayIndexOutOfBoundsException

correct: 4

reason: ArrayIndexOutOfBoundsException is a built in exception that is caused when we try to access an index location which is beyond the length of an array.

Q148What is the output of this program?  
 class exception\_handling   
 {  
 public static void main(String args[])   
 {  
 try   
 {  
 System.out.print("A");  
 throw new NullPointerException ("Hello");  
 }  
 catch(ArithmeticException e)   
 {  
 System.out.print("B");   
 }  
 }  
 }

1. A

2. B

3. Hello

4. Runtime Exception

correct: 3

reason: Because we can’t go beyond array limit

Q149What is the output of this program?  
class exception\_handling   
 {  
 public static void main(String args[])   
 {  
 try   
 {  
 int i, sum;  
 sum = 10;  
 for (i = -1; i < 3 ;++i)   
 {  
 sum = (sum / i);  
 System.out.print(i);  
 }  
 }  
 catch(ArithmeticException e)   
 {   
 System.out.print("0");  
 }  
 }  
 }

1. -1

2. 0

3. -10

4. -101

correct: 3

reason: For the 1st iteration -1 is displayed. The 2nd exception is caught in catch block and 0 is displayed.

Q150What exception thrown by parseInt() method?

1. ArithmeticException

2. ClassNotFoundException

3. NullPointerException

4. NumberFormatException

correct: 4

reason: parseInt() method parses input into integer. The exception thrown by this method is NumberFormatException.

Q151What is the output of this program?  
class exception\_handling   
 {  
 public static void main(String args[])   
 {  
 try   
 {  
 System.out.print("Hello" + " " + 1 / 0);  
 }  
 catch(ArithmeticException e)   
 {  
 System.out.print("World");   
 }  
 }  
 }

1. Hello

2. World

3. Hello World

4. HelloWorld

correct: 2

reason: System.ou.print() function first converts the whole parameters into a string and then prints, before “Hello” goes to output stream 1 / 0 error is encountered which is cached by catch block printing just “World”.

Q152What is multithreaded programming?

1. It’s a process in which two different processes run simultaneously

2. It’s a process in which two or more parts of same process run simultaneously

3. It’s a process in which many different process are able to access same information

4. It’s a process in which a single process can access information from many sources

correct: 2

reason: Multithreaded programming a process in which two or more parts of the same process run simultaneously.

Q153What decides thread priority?

1. process

2. process scheduler

3. Thread

4. Thread Scheduler

correct: 4

reason: Thread scheduler decides the priority of the thread execution. This cannot guarantee that higher priority thread will be executed first, it depends on thread scheduler implementation that is OS dependent.

Q154What requires less resources?

1. Thread

2. process

3. Thread and Process

4. Neither Thread nor Process

correct: 1

reason: Thread is a lightweight and requires less resources to create and exist in the process. Thread shares the process resources.

Q155Which of these method can be used to make the main thread to be executed last among all the threads?

1. stop()

2. sleep()

3. join()

4. call()

correct: 2

reason: By calling sleep() within main(), with long enough delay to ensure that all child threads terminate prior to the main thread.

Q156Which of these method is used to find out that a thread is still running or not?

1. run()

2. Alive()

3. isAlive()

4. checkRun()

correct: 3

reason: The isAlive( ) method returns true if the thread upon which it is called is still running. It returns false otherwise.

Q157What is the default value of priority variable MIN\_PRIORITY AND MAX\_PRIORITY?

1. 0 & 256

2. 0 & 1

3. 1 & 10

4. 1 & 256

correct: 3

reason: The priority dat type is Integers and it is from 1 – 10

Q158Which of these method waits for the thread to terminate?

1. sleep()

2. isAlive()

3. join()

4. stop()

correct: 3

reason: java.lang.Thread class provides the join() method which allows one thread to wait until another thread completes its execution.

Q159Which of these method is used to explicitly set the priority of a thread?

1. set()

2. make()

3. setPriority()

4. makePriority()

correct: 3

reason: The default value of priority given to a thread is 5 but we can explicitly change that value between the permitted values 1 & 10, this is done by using the method setPriority().

Q160Which of these method is used to implement Runnable interface?

1. stop()

2. run()

3. runThread()

4. stopThread()

correct: 2

reason: To implement Runnable interface, a class needs only to implement a single method called run().

Q161Which of these interface is implemented by Thread class?

1. Runnable

2. Connections

3. set

4. MapConnections

correct: 1

reason: Runnable interface is used to create thread.

Q162Which of these method is used to begin the execution of a thread?

1. run()

2. start()

3. runThread()

4. startThread()

correct: 2

reason: The start() method of thread class is used to begin the execution of thread.

Q163How can a thread go from waiting to Runnable state?

1. notify/notifyAll

2. when sleep time is up

3. using resume(0 method when thread was suspended

4. All

correct: 4

reason: A thread in the runnable state is executing in the Java virtual machine but it may be waiting for other resources from the operating system such as processor.

Q164Which of these class is used to make a thread?

1. String

2. System

3. Thread

4. Runnable

correct: 3

reason: Thread class is used to make threads in java, Thread encapsulates a thread of execution. To create a new thread the program will either extend Thread or implement the Runnable interface.

Q165Which of these method of Thread class is used to find out the priority given to a thread?

1. get()

2. ThreadPriority()

3. getPriority()

4. getThreadPriority()

correct: 3

reason: getPriority() Method is used to find priority of the thread.

Q166What does not prevent JVM from terminating?

1. process

2. Daemon Thread

3. user Thread

4. JVM Thread

correct: 2

reason: Daemon thread runs in the background and does not prevent JVM from terminating. Child of daemon thread is also daemon thread.

Q167Which of these statements is incorrect?

1. A thread can be formed by implementing Runnable interface only.

2. A thread can be formed by a class that extends Thread class.

3. start() method is used to begin execution of the thread.

4. run() method is used to begin execution of a thread before start() method in special cases.

correct: 4

reason: run() method is used to define the code that constitutes the new thread, it contains the code to be executed. start() method is used to begin execution of the thread that is execution of run(). run() itself is never used for starting execution of the thread.

Q168What is synchronization in reference to a thread?

1. It’s a process of handling situations when two or more threads need access to a shared resource.

2. Its a process by which many thread are able to access same shared resource simultaneously.

3. Its a process by which a method is able to access many different threads simultaneously.

4. Its a method that allow to many threads to access any information require.

correct: 1

reason: When two or more threads need to access the same shared resource, they need some way to ensure that the resource will be used by only one thread at a time, the process by which this is achieved is called synchronization.

Q169Which of the following is a correct constructor for thread?

1. Thread(Runnable a, String str)

2. Thread(int priority)

3. Thread(Runnable a, int priority)

4. Thread(Runnable a, ThreadGroup t)

correct: 1

reason: Thread(Runnable a, String str) is a valid constructor for thread. Thread() is also a valid constructor.

Q170Which of the following stops execution of a thread?

1. Calling SetPriority() method on a Thread object

2. Calling notify() method on an object

3. Calling wait() method on an object

4. Calling read() method on an InputStream object

correct: 2

reason: notify() wakes up a single thread which is waiting for this object.

Q171Which of these method is used to tell the calling thread to give up a monitor and go to sleep until some other thread enters the same monitor?

1. wait()

2. notify()

3. notifyAll()

4. sleep()

correct: 1

reason: wait() method is used to tell the calling thread to give up a monitor and go to sleep until some other thread enters the same monitor. This helps in avoiding polling and minimizes CPU idle time.

Q172Which of this method is used to avoid polling in Java?

1. wait()

2. notify()

3. notifyAll()

4. All of the mentioned above

correct: 4

reason: Polling is a usually implemented by looping in CPU is wastes CPU time, one thread being executed depends on other thread output and the other thread depends on the response on the data given to the first thread. In such situation CPU time is wasted, in Java this is avoided by using methods wait(), notify() and notifyAll().

Q173Which of these method wakes up all the threads?

1. wakeAll()

2. notify()

3. start()

4. notifyAll()

correct: 4

reason: notifyAll() wakes up all the threads that called wait() on the same object. The highest priority thread will run first.

Q174Which of the following will ensure the thread will be in running state?

1. yield()

2. notify()

3. wait()

4. Thread.killThread()

correct: 3

reason: wait() always causes the current thread to go into the object’s wait pool. Hence, using this in a thread will keep it in running state.

Q175What is true about threading?

1. run() method calls start() method and runs the code

2. run() method creates new thread

3. run() method can be called directly without start() method being called

4. start() method creates new thread and calls code written in run() method

correct: 4

reason: start() eventually calls run() method. Start() method creates thread and calls the code written inside run method.

Q176What is the output of this program?  
 class multithreaded-programing  
 {  
 public static void main(String args[])  
 {  
 Thread t = Thread.currentThread();  
 System.out.println(t);   
 }  
 }

1. Thread[5,main]

2. Thread[main,0]

3. Thread[main,5]

4. Thread[main,5,main]

correct: 4

reason: 1. main is name of the current thread  
2.priority of the thread if not mentioned it will have default priority which is 5  
3.Thread group name.

Q177What should not be done to avoid deadlock?

1. Avoid using multiple threads

2. Avoid hold several locks at once

3. Execute foreign code while holding a lock

4. Use interruptible locks

correct: 3

reason: To avoid deadlock situation in Java programming do not execute foreign code while holding a lock.

Q178What is true about time slicing?

1. Time slicing is OS service that allocates CPU time to available runnable thread

2. Time slicing is the process to divide the available CPU time to available runnable thread

3. Time slicing depends on its implementation in OS

4. Time slicing allocates more resources to thread

correct: 2

reason: Time slicing is the process to divide the available CPU time to available runnable thread.

Q179Which of these statements is incorrect?

1. By multithreading CPU idle time is minimized, and we can take maximum use of it

2. By multitasking CPU idle time is minimized, and we can take maximum use of it

3. Two thread in Java can have the same priority

4. A thread can exist only in two states, running and blocked

correct: 4

reason: Thread exist in several states, a thread can be running, suspended, blocked, terminated & ready to run.

Q180What will happen if two thread of the same priority are called to be processed simultaneously?

1. Anyone will be executed first lexographically

2. Both of them will be executed simultaneously

3. None of them will be executed

4. It is dependent on the operating system

correct: 4

reason: In cases where two or more thread with same priority are competing for CPU cycles, different operating system handle this situation differently. Some execute them in time sliced manner some depending on the thread they call.

Q181What is the output of this program?  
 class newthread extends Thread  
 {  
 Thread t;  
 String name;  
 newthread(String threadname)  
 {  
 name = threadname;  
 t = new Thread(this,name);  
 t.start();  
 }  
 public void run()  
 {  
 }  
 }  
 class multithreaded\_programing  
 {  
 public static void main(String args[])  
 {  
 newthread obj1 = new newthread("one");  
 newthread obj2 = new newthread("two");  
 try  
 {  
 Thread.sleep(1000);   
 System.out.print(obj1.t.isAlive());  
 }  
 catch(InterruptedException e)  
 {  
 System.out.print("Main thread interrupted");  
 }  
 }  
 }

1. TRUE

2. FALSE

3. main Thread Interrupted

4. None of the mentioned

correct: 2

reason: Thread.sleep(1000) has caused all the threads to be suspended for some time, hence onj1.t.isAlive() returns false.