

java-questions-answers-freshers-experienced

1. What is the range of short data type in Java?

- a) -128 to 127
- b) -32768 to 32767
- c) -2147483648 to 2147483647
- d) None of the mentioned

Answer: b

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

2. What is the range of byte data type in Java?

- a) -128 to 127
- b) -32768 to 32767
- c) -2147483648 to 2147483647
- d) None of the mentioned

Answer: a

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

3. Which of the following are legal lines of Java code?

```
1. int w = (int)888.8;
2. byte x = (byte)100L;
3. long y = (byte)100;
4. byte z = (byte)100L;
```

- a) 1 and 2
- b) 2 and 3
- c) 3 and 4
- d) All statements are correct

Answer: d

Explanation: Statements (1), (2), (3), and (4) are correct. (1) is correct because when a floating-point number (a double in this case) is cast to an int, it simply loses the digits after the decimal. (2) and (4) are correct because a long can be cast into a byte. If the long is over 127, it loses its most significant (leftmost) bits. (3) actually works, even though a cast is not necessary, because a long can store a byte.

4. An expression involving byte, int, and literal numbers is promoted to which of these?

- a) int
- b) long
- c) byte
- d) float

Answer: a

Explanation: An expression involving bytes, ints, shorts, literal numbers, the entire expression is promoted to int before any calculation is done.

5. Which of these literals can be contained in float data type variable?

- a) -1.7e+308
- b) -3.4e+038
- c) +1.7e+308
- d) -3.4e+050

Answer: b

Explanation: Range of float data type is $-(3.4e38)$ To $+(3.4e38)$

6. Which data type value is returned by all transcendental math functions?

- a) int
- b) float
- c) double
- d) long

Answer: c

Explanation: None.

7. What will be the output of the following Java code?

```
1.      class average {
2.          public static void main(String args[])
3.          {
4.              double num[] = {5.5, 10.1, 11, 12.8, 56.9, 2.5};
5.              double result;
6.              result = 0;
7.              for (int i = 0; i < 6; ++i)
8.                  result = result + num[i];
9.              System.out.print(result/6);
10.
11.          }
12.      }
```

- a) 16.34
- b) 16.5666666644
- c) 16.466666666666667
- d) 16.466666666666666

Answer: c

Explanation: None.

output:

```
$ javac average.java
$ java average
16.466666666666667
```

8. What will be the output of the following Java statement?

```
1. class output {
2.     public static void main(String args[])
3.     {
4.         double a, b,c;
5.         a = 3.0/0;
6.         b = 0/4.0;
7.         c=0/0.0;
8.     }
```

```
9.          System.out.println(a);
10.         System.out.println(b);
11.         System.out.println(c);
12.     }
13. }
```

- a) Infinity
- b) 0.0
- c) NaN
- d) all of the mentioned

Answer: d

Explanation: For floating point literals, we have constant value to represent (10/0.0) infinity either positive or negative and also have NaN (not a number for undefined like 0/0.0), but for the integral type, we don't have any constant that's why we get an arithmetic exception.

9. What will be the output of the following Java code?

```
1.  class increment {
2.      public static void main(String args[])
3.      {
4.          int g = 3;
5.          System.out.print(++g * 8);
6.      }
7.  }
```

- a) 25
- b) 24
- c) 32
- d) 33

Answer: c

Explanation: Operator ++ has more preference than *, thus g becomes 4 and when multiplied by 8 gives 32.
output:

```
$ javac increment.java
$ java increment
32
```

10. What will be the output of the following Java code?

```
1.  class area {
2.      public static void main(String args[])
3.      {
4.          double r, pi, a;
5.          r = 9.8;
6.          pi = 3.14;
```

```
7.          a = pi * r * r;  
8.          System.out.println(a);  
9.      }  
10. }
```

- a) 301.5656
- b) 301
- c) 301.56
- d) 301.56560000

Answer: a

Explanation: None.

output:

```
$ javac area.java  
$ java area  
301.5656
```

1. What is the numerical range of a char data type in Java?

- a) -128 to 127
- b) 0 to 256
- c) 0 to 32767
- d) 0 to 65535

Answer: d

Explanation: Char occupies 16-bit in memory, so it supports 2^{16} i.e from 0 to 65535.

2. Which of these coding types is used for data type characters in Java?

- a) ASCII
- b) ISO-LATIN-1
- c) UNICODE
- d) None of the mentioned

Answer: c

Explanation: Unicode defines fully international character set that can represent all the characters found in all human languages. Its range is from 0 to 65536.

3. Which of these values can a boolean variable contain?

- a) True & False
- b) 0 & 1
- c) Any integer value
- d) true

Answer: a

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

4. Which of these occupy first 0 to 127 in Unicode character set used for characters in Java?

- a) ASCII
- b) ISO-LATIN-1
- c) None of the mentioned
- d) ASCII and ISO-LATIN1

Answer: d

Explanation: First 0 to 127 character set in Unicode are same as those of ISO-LATIN-1 and ASCII.

5. Which one is a valid declaration of a boolean?

- a) `boolean b1 = 1;`
- b) `boolean b2 = 'false';`
- c) `boolean b3 = false;`
- d) `boolean b4 = 'true'`

Answer: c
Explanation: Boolean can only be assigned true or false literals.

6. What will be the output of the following Java program?

```
1.      class array_output {
2.          public static void main(String args[])
3.          {
4.              char array_variable [] = new char[10];
5.              for (int i = 0; i < 10; ++i) {
6.                  array_variable[i] = 'i';
7.                  System.out.print(array_variable[i] + " " );
8.                  i++;
9.              }
10.         }
11.     }
```

- a) `iiiii`
- b) `0 1 2 3 4`
- c) `ijklm`
- d) None of the mentioned

Answer: a
Explanation: None.
output:

```
$ javac array_output.java
$ java array_output
i i i i i
```

7. What will be the output of the following Java program?

```
1.      class mainclass {
2.          public static void main(String args[])
3.          {
4.              char a = 'A';
5.              a++;
6.              System.out.print((int)a);
7.          }
8.     }
```

- a) 66

- b) 67
- c) 65
- d) 64

Answer: a

Explanation: ASCII value of 'A' is 65, on using ++ operator character value increments by one.
output:

```
$ javac mainclass.java
$ java mainclass
66
```

8. What will be the output of the following Java program?

```
1.      class mainclass {
2.          public static void main(String args[])
3.          {
4.              boolean var1 = true;
5.              boolean var2 = false;
6.              if (var1)
7.                  System.out.println(var1);
8.              else
9.                  System.out.println(var2);
10.         }
11.     }
```

- a) 0
- b) 1
- c) true
- d) false

Answer: c

Explanation: None.

output:

```
$ javac mainclass.java
$ java mainclass
true
```

9. What will be the output of the following Java code?

```
1.      class booloperators {
2.          public static void main(String args[])
3.          {
4.              boolean var1 = true;
5.              boolean var2 = false;
6.              System.out.println((var1 & var2));
7.          }
```

```
8.      }
```

- a) 0
- b) 1
- c) true
- d) false

Answer: d

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

output:

```
$ javac booloperators.java
$ java booloperators
false
```

10. What will be the output of the following Java code?

```
1.      class asciicodes {
2.
3.          public static void main(String args[])
4.          {
5.
6.              char var1 = 'A';
7.              char var2 = 'a';
8.
9.              System.out.println((int)var1 + " " + (int)var2);
10.         }
11.     }
```

- a) 162
- b) 65 97
- c) 67 95
- d) 66 98

Answer: b

Explanation: ASCII code for 'A' is 65 and for 'a' is 97.

output:

```
$ javac asciicodes.java
$ java asciicodes
65 97
```

1. What is the order of variables in Enum?

- a) Ascending order
- b) Descending order
- c) Random order
- d) Depends on the order() method

Answer: a

Explanation: The compareTo() method is implemented to order the variable in ascending order.

Answer: b
Explanation: Enum does not have a public constructor.

3. What will be the output of the following Java code?

```
1.      enum Season
2.      {
```

```
3.         WINTER, SPRING, SUMMER, FALL
4.     };
5.     System.out.println(Season.WINTER.ordinal());
```

- a) 0
- b) 1
- c) 2
- d) 3

Answer: a

Explanation: ordinal() method provides number to the variables defined in Enum.

4. If we try to add Enum constants to a TreeSet, what sorting order will it use?

- a) Sorted in the order of declaration of Enums
- b) Sorted in alphabetical order of Enums
- c) Sorted based on order() method
- d) Sorted in descending order of names of Enums

Answer: a

Explanation: Tree Set will sort the values in the order in which Enum constants are declared.

5. What will be the output of the following Java code snippet?

```
1. class A
2. {
3.
4. }
5.
6. enum Enums extends A
7. {
8.     ABC, BCD, CDE, DEF;
9. }
```

- a) Runtime Error
- b) Compilation Error
- c) It runs successfully
- d) EnumNotDefined Exception

Answer: b

Explanation: Enum types cannot extend class.

6. What will be the output of the following Java code snippet?

```
1. enum Levels
2. {
3.     private TOP,
4.
5.     public MEDIUM,
```



```
6.  
7.     protected BOTTOM;  
8. }
```

- a) Runtime Error
- b) EnumNotDefined Exception
- c) It runs successfully
- d) Compilation Error

Answer: d

Explanation: Enum cannot have any modifiers. They are public, static and final by default.

7. What will be the output of the following Java code snippet?

```
1. enum Enums  
2. {  
3.     A, B, C;  
4.  
5.     private Enums()  
6.     {  
7.         System.out.println(10);  
8.     }  
9. }  
10.  
11. public class MainClass  
12. {  
13.     public static void main(String[] args)  
14.     {  
15.         Enum en = Enums.B;  
16.     }  
17. }
```

Answer: a

Explanation: The constructor of Enums is called which prints 10.

8. Which method returns the elements of Enum class?

- a) getEnums()
- b) getEnumConstants()
- c) getEnumList()
- d) getEnum()

Answer: b

Explanation: getEnumConstants() returns the elements of this enum class or null if this Class object does not represent an enum type.

9. Which class does all the Enums extend?

- a) Object
- b) Enums
- c) Enum
- d) EnumClass

Answer: c

Explanation: All enums implicitly extend `java.lang.Enum`. Since Java does not support multiple inheritance, an enum cannot extend anything else. Answer: a

Explanation: Enums are type-safe as they have own name-space.

1. Which of the following is the advantage of BigDecimal over double?

- a) Syntax
- b) Memory usage
- c) Garbage creation
- d) Precision

Answer: d

Explanation: `BigDecimal` has unnatural syntax, needs more memory and creates a great amount of garbage. But it has a high precision which is useful for some calculations like money.

2. Which of the below data type doesn't support overloaded methods for +,-,* and /?

- a) int
- b) float
- c) double
- d) `BigDecimal`

Answer: d

Explanation: `int`, `float`, `double` provide overloaded methods for +,-,* and /. `BigDecimal` does not provide these overloaded methods.

3. What will be the output of the following Java code snippet?

```
1.    double a = 0.02;
2.    double b = 0.03;
3.    double c = b - a;
4.    System.out.println(c);
5.
6.    BigDecimal _a = new BigDecimal("0.02");
7.    BigDecimal _b = new BigDecimal("0.03");
8.    BigDecimal _c = b.subtract(_a);
9.    System.out.println(_c);
```

4. What is the base of BigDecimal data type?

- a) Base 2
- b) Base 8
- c) Base 10
- d) Base e

Answer: c

Explanation: A `BigDecimal` is $n \cdot 10^{\text{scale}}$ where n is an arbitrary large signed integer. Scale can be thought of as the number of digits to move the decimal point to left or right.

5. What is the limitation of toString() method of BigDecimal?

- a) There is no limitation
- b) toString returns null
- c) toString returns the number in expanded form
- d) toString uses scientific notation

Answer: d
Explanation: toString() of BigDecimal uses scientific notation to represent numbers known as canonical representation. We must use toPlainString() to avoid scientific notation.

6. Which of the following is not provided by BigDecimal?

- a) scale manipulation
- b) + operator
- c) rounding
- d) hashing

Answer: b
Explanation: toBigInteger() converts BigDecimal to a BigInteger.toBigIntegerExact() converts this BigDecimal to a BigInteger by checking for lost information.

7. BigDecimal is a part of which package?

- a) java.lang
- b) java.math
- c) java.util
- d) java.io

Answer: b
Explanation: BigDecimal is a part of java.math. This package provides various classes for storing numbers and mathematical operations.

8. What is BigDecimal.ONE?

- a) wrong statement
- b) custom defined statement
- c) static variable with value 1 on scale 10
- d) static variable with value 1 on scale 0

Answer: d
Explanation: BigDecimal.ONE is a static variable of BigDecimal class with value 1 on scale 0.

9. Which class is a library of functions to perform arithmetic operations of BigInteger and BigDecimal?

- a) MathContext
- b) MathLib
- c) BigLib
- d) BigContext

Answer: a
Explanation: MathContext class is a library of functions to perform arithmetic operations of BigInteger and BigDecimal.

10. What will be the output of the following Java code snippet?

```
1. public class AddDemo
2. {
3.     public static void main(String args[])
4.     {
5.         BigDecimal b = new BigDecimal("23.43");
```

```

6.         BigDecimal br = new BigDecimal("24");
7.
8.         BigDecimal bres = b.add(new BigDecimal("450.23"));
9.
10.        System.out.println("Add: "+bres);
11.
12.
13.        MathContext mc = new MathContext(2, RoundingMode.DOWN);
14.        BigDecimal bdecMath = b.add(new BigDecimal("450.23"), mc);
15.        System.out.println("Add using MathContext: "+bdecMath);
16.    }
17. }

```

Answer: b

Explanation: add() adds the two numbers, MathContext provides library for carrying out various arithmetic operations.

1. How to format date from one form to another?

- a) SimpleDateFormat**
- b) DateFormat**
- c) SimpleFormat**
- d) DateConverter**

Answer: a

Explanation: SimpleDateFormat can be used as

```

Date now = new Date();
SimpleDateFormat sdf = new SimpleDateFormat ("yyyy-mm-dd'T'hh:MM:ss");
String nowStr = sdf.format(now);
System.out.println("Current Date: " + );

```

Answer: b

Explanation: SimpleDateFormat is not thread safe. In the multithreaded environment, we need to manage threads explicitly.

5. How to identify if a timezone is eligible for DayLight Saving?

- a) useDaylightTime() of Time class**
- b) useDaylightTime() of Date class**
- c) useDaylightTime() of TimeZone class**
- d) useDaylightTime() of DateTime class**

Answer: c

Explanation: public abstract boolean useDaylightTime() is provided in TimeZone class.

6. What is the replacement of joda time library in java 8?

- a) java.time (JSR-310)**
- b) java.date (JSR-310)**
- c) java.joda**
- d) java.jodaTime**

Answer: a

Explanation: In java 8, we are asked to migrate to java.time (JSR-310) which is a core part of the JDK which replaces joda library project.

7. How is Date stored in database?

- a) java.sql.Date**
- b) java.util.Date**

- c) `java.sql.Date`
- d) `java.util.Date`

Answer: a

Explanation: `java.sql.Date` is the datatype of Date stored in database.

8. What does `LocalTime` represent?

- a) Date without time
- b) Time without Date
- c) Date and Time
- d) Date and Time with timezone

Answer: b

Explanation: `LocalTime` of `joda` library represents time without date.

9. How to get difference between two dates?

- a) `long diffInMilli = java.time.Duration.between(dateTime1, dateTime2).toMillis();`
- b) `long diffInMilli = java.time.difference(dateTime1, dateTime2).toMillis();`
- c) `Date diffInMilli = java.time.Duration.between(dateTime1, dateTime2).toMillis();`
- d) `Time diffInMilli = java.time.Duration.between(dateTime1, dateTime2).toMillis();`

Answer: a

Explanation: Java 8 provides a method called `between` which provides `Duration` between two times.

10. How to get UTC time?

- a) `Time.getUTC();`
- b) `Date.getUTC();`
- c) `Instant.now();`
- d) `TimeZone.getUTC();`

Answer: c

Explanation: In java 8, `Instant.now()` provides current time in UTC/GMT.

1. Which of these is long data type literal?

- a) `0x99ffL`
- b) `ABCDEFG`
- c) `0x99ffa`
- d) `99671246`

Answer: a

Explanation: Data type long literals are appended by an upper or lowercase L. `0x99ffL` is hexadecimal long literal.

2. Which of these can be returned by the operator `&`?

- a) Integer
- b) Boolean
- c) Character
- d) Integer or Boolean

Answer: d

Explanation: We can use binary ampersand operator on integers/chars (and it returns an integer) or on booleans (and it returns a boolean).

3. Literals in java must be appended by which of these?

- a) L
- b) l
- c) D
- d) L and I

Answer: d

Explanation: Data type long literals are appended by an upper or lowercase L.

4. Literal can be of which of these data types?

- a) integer**
- b) float**
- c) boolean**
- d) all of the mentioned**

Answer: d

Explanation: None

5. Which of these can not be used for a variable name in Java?

- a) identifier**
- b) keyword**
- c) identifier & keyword**
- d) none of the mentioned**

Answer: b

Explanation: Keywords are specially reserved words which can not be used for naming a user defined variable, example: class, int, for etc.

6. What will be the output of the following Java program?

```
1.      class evaluate
2.      {
3.          public static void main(String args[])
4.          {
5.              int a[] = {1,2,3,4,5};
6.              int d[] = a;
7.              int sum = 0;
8.              for (int j = 0; j < 3; ++j)
9.                  sum += (a[j] * d[j + 1]) + (a[j + 1] * d[j]);
10.             System.out.println(sum);
11.         }
12.     }
```

- a) 38**
- b) 39**
- c) 40**
- d) 41**

Answer: c

Explanation: None.

output:

```
$ javac evaluate.java
$ java evaluate
40
```

7. What will be the output of the following Java program?

```

1.     class array_output
2.     {
3.         public static void main(String args[])
4.         {
5.             int array_variable [] = new int[10];
6.             for (int i = 0; i < 10; ++i) {
7.                 array_variable[i] = i/2;
8.                 array_variable[i]++;
9.                 System.out.print(array_variable[i] + " ");
10.                i++;
11.            }
12.
13.        }
14.    }

```

a) 0 2 4 6 8

b) 1 2 3 4 5

c) 0 1 2 3 4 5 6 7 8 9

d) 1 2 3 4 5 6 7 8 9 10

Answer: b

Explanation: When an array is declared using new operator then all of its elements are initialized to 0 automatically. for loop body is executed 5 times as whenever controls comes in the loop i value is incremented twice, first by i++ in body of loop then by ++i in increment condition of for loop.

output:

```

$ javac array_output.java
$ java array_output
1 2 3 4 5

```

8. What will be the output of the following Java program?

```

1.     class variable_scope
2.     {
3.         public static void main(String args[])
4.         {
5.             int x;
6.             x = 5;
7.             {
8.                 int y = 6;
9.                 System.out.print(x + " " + y);
10.            }

```

```
11.         System.out.println(x + " " + y);
12.     }
13. }
```

- a) 5 6 5 6
- b) 5 6 5
- c) Runtime error
- d) Compilation error

Answer: d

Explanation: Second print statement doesn't have access to y , scope y was limited to the block defined after initialization of x.

output:

```
$ javac variable_scope.java
Exception in thread "main" java.lang.Error: Unresolved compilation problem: y cannot be resolved t
```

9. Which of these is an incorrect string literal?

- a) “Hello World”
- b) “Hello\nWorld”
- c) “\”Hello World\””
- d)

10. What will be the output of the following Java program?

```
1.     class dynamic_initialization
2.     {
3.         public static void main(String args[])
4.         {
5.             double a, b;
6.             a = 3.0;
7.             b = 4.0;
8.             double c = Math.sqrt(a * a + b * b);
9.             System.out.println(c);
10.        }
11.    }
```

- a) 5.0
- b) 25.0
- c) 7.0
- d) Compilation Error

Answer: a

*Explanation: Variable c has been dynamically initialized to square root of $a * a + b * b$, during run time.*

output:

```
$ javac dynamic_initialization.java
$ java dynamic_initialization
5.0
```

1. Which of these is necessary condition for automatic type conversion in Java?

- a) The destination type is smaller than source type
- b) The destination type is larger than source type
- c) The destination type can be larger or smaller than source type
- d) None of the mentioned

Answer: b

Explanation: None.

2. What is the prototype of the default constructor of this Java class?

```
public class prototype { }
```

- a) prototype()
- b) prototype(void)
- c) public prototype(void)
- d) public prototype()

Answer: d

Explanation: None.

3. What will be the error in the following Java code?

```
byte b = 50;  
b = b * 50;
```

- a) b cannot contain value 100, limited by its range
- b) * operator has converted b * 50 into int, which can not be converted to byte without casting
- c) b cannot contain value 50
- d) No error in this code

Answer: b

Explanation: While evaluating an expression containing int, bytes or shorts, the whole expression is converted to int then evaluated and the result is also of type int.

4. If an expression contains double, int, float, long, then the whole expression will be promoted into which of these data types?

- a) long
- b) int
- c) double
- d) float

Answer: c

Explanation: If any operand is double the result of an expression is double.

5. What is Truncation in Java?

- a) Floating-point value assigned to an integer type
- b) Integer value assigned to floating type
- c) Floating-point value assigned to an Floating type
- d) Integer value assigned to floating type

Answer: a

Explanation: None.

6. What will be the output of the following Java code?

```
1.      class char_increment  
2.      {  
3.          public static void main(String args[])
```

```

4.      {
5.          char c1 = 'D';
6.          char c2 = 84;
7.          c2++;
8.          c1++;
9.          System.out.println(c1 + " " + c2);
10.     }
11. }

```

- a) E U
- b) U E
- c) V E
- d) U F

Answer: a

Explanation: 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.

output:

```

$ javac char_increment.java
$ java char_increment
E U

```

7. What will be the output of the following Java code?

```

1.      class conversion
2.      {
3.          public static void main(String args[])
4.          {
5.              double a = 295.04;
6.              int b = 300;
7.              byte c = (byte) a;
8.              byte d = (byte) b;
9.              System.out.println(c + " " + d);
10.         }
11.     }

```

- a) 38 43
- b) 39 44
- c) 295 300
- d) 295.04 300

Answer: b

Explanation: 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.

output:

```
$ javac conversion.java
$ java conversion
39 44
```

8. What will be the output of the following Java code?

```
1.      class A
2.      {
3.          final public int calculate(int a, int b) { return 1; }
4.      }
5.      class B extends A
6.      {
7.          public int calculate(int a, int b) { return 2; }
8.      }
9.      public class output
10.     {
11.         public static void main(String args[])
12.         {
13.             B object = new B();
14.             System.out.print("b is " + b.calculate(0, 1));
15.         }
16.     }
```

- a) b is : 2
- b) b is : 1
- c) Compilation Error
- d) An exception is thrown at runtime

Answer: c

Explanation: The code does not compile because the method `calculate()` in class *A* is *final* and so cannot be overridden by method of class *b*.

9. What will be the output of the following Java program, if we run as “java main_arguments 1 2 3”?

```
1.      class main_arguments
2.      {
3.          public static void main(String [] args)
4.          {
5.              String [][] argument = new String[2][2];
6.              int x;
7.              argument[0] = args;
```

```

8.            x = argument[0].length;

9.            for (int y = 0; y < x; y++)

10.           System.out.print(" " + argument[0][y]);

11.        }

12.    }

```

- a) 1 1
- b) 1 0
- c) 1 0 3
- d) 1 2 3

Answer: d

Explanation: In `argument[0] = args;`, the reference variable `arg[0]`, which was referring to an array with two elements, is reassigned to an array (`args`) with three elements.

Output:

```

$ javac main_arguments.java
$ java main_arguments
1 2 3

```

10. What will be the output of the following Java program?

```

1.    class c

2.    {

3.        public void main( String[] args )

4.        {

5.            System.out.println( "Hello" + args[0] );

6.        }

7.    }

```

- a) Hello c
- b) Hello
- c) Hello world
- d) Runtime Error

Answer: d

Explanation: A runtime error will occur owing to the main method of the code fragment not being declared static.

Output:

```

$ javac c.java
Exception in thread "main" java.lang.NoSuchMethodError: main

```

1. Which of these operators is used to allocate memory to array variable in Java?

- a) malloc
- b) alloc
- c) new
- d) new malloc

Answer: c

Explanation: Operator `new` allocates a block of memory specified by the size of an array, and gives the reference of memory allocated to the array variable.

2. Which of these is an incorrect array declaration?

- a) `int arr[] = new int[5]`
- b) `int [] arr = new int[5]`
- c) `int arr[] = new int[5]`
- d) `int arr[] = int [5] new`

Answer: d

Explanation: Operator new must be succeeded by array type and array size.

3. What will be the output of the following Java code?

```
int arr[] = new int [5];  
System.out.print(arr);
```

- a) 0
- b) value stored in arr[0]
- c) 00000
- d) Class [\[email protected\]](#) hashCode in hexadecimal form

Answer: d

Explanation: If we trying to print any reference variable internally, `toString()` will be called which is implemented to return the String in following form:

[\[email protected\]](#) in hexadecimal form

4. Which of these is an incorrect Statement?

- a) It is necessary to use new operator to initialize an array
- b) Array can be initialized using comma separated expressions surrounded by curly braces
- c) Array can be initialized when they are declared
- d) None of the mentioned

Answer: a

Explanation: Array can be initialized using both new and comma separated expressions surrounded by curly braces
example : `int arr[5] = new int[5];` and `int arr[] = { 0, 1, 2, 3, 4};`

5. Which of these is necessary to specify at time of array initialization?

- a) Row
- b) Column
- c) Both Row and Column
- d) None of the mentioned

Answer: a

Explanation: None.

6. What will be the output of the following Java code?

```
1.    class array_output  
2.    {  
3.        public static void main(String args[])  
4.        {  
5.            int array_variable [] = new int[10];  
6.            for (int i = 0; i < 10; ++i)  
7.            {  
8.                array_variable[i] = i;
```

```

9.             System.out.print(array_variable[i] + " ");
10.            i++;
11.        }
12.    }
13. }

```

a) 0 2 4 6 8

b) 1 3 5 7 9

c) 0 1 2 3 4 5 6 7 8 9

d) 1 2 3 4 5 6 7 8 9 10

Answer: a

Explanation: When an array is declared using new operator then all of its elements are initialized to 0 automatically. for loop body is executed 5 times as whenever controls comes in the loop i value is incremented twice, first by i++ in body of loop then by ++i in increment condition of for loop.

output:

```

$ javac array_output.java
$ java array_output
0 2 4 6 8

```

7. What will be the output of the following Java code?

```

1.    class multidimention_array
2.    {
3.        public static void main(String args[])
4.        {
5.            int arr[][] = new int[3][];
6.            arr[0] = new int[1];
7.            arr[1] = new int[2];
8.            arr[2] = new int[3];
9.            int sum = 0;
10.           for (int i = 0; i < 3; ++i)
11.               for (int j = 0; j < i + 1; ++j)
12.                   arr[i][j] = j + 1;
13.           for (int i = 0; i < 3; ++i)
14.               for (int j = 0; j < i + 1; ++j)
15.                   sum + = arr[i][j];
16.           System.out.print(sum);
17.       }
18.   }

```

a) 11

b) 10

c) 13
d) 14

Answer: b
Explanation: `arr[][]` is a 2D array, array has been allotted memory in parts. 1st row contains 1 element, 2nd row contains 2 elements and 3rd row contains 3 elements. each element of array is given $i + j$ value in loop. sum contains addition of all the elements of the array.
output:

```
$ javac multidimention_array.java
$ java multidimention_array
10
```

8. What will be the output of the following Java code?

```
1.      class evaluate
2.      {
3.          public static void main(String args[])
4.          {
5.              int arr[] = new int[] {0 , 1, 2, 3, 4, 5, 6, 7, 8, 9};
6.              int n = 6;
7.              n = arr[arr[n] / 2];
8.              System.out.println(arr[n] / 2);
9.          }
10.     }
```

a) 3
b) 0
c) 6
d) 1

Answer: d
Explanation: Array `arr` contains 10 elements. `n` contains 6 thus in next line `n` is given value 3 printing `arr[3]/2` i.e $3/2 = 1$ because of int Value, by int values there is no rest. If this values would be float the result would be 1.5.
output:

```
$ javac evaluate.java
$ java evaluate
1
```

9. What will be the output of the following Java code?

```
1.      class array_output
2.      {
3.          public static void main(String args[])
4.          {
5.              char array_variable [] = new char[10];
6.              for (int i = 0; i < 10; ++i)
7.              {
```

```

8.         array_variable[i] = 'i';
9.         System.out.print(array_variable[i] + "");
10.    }
11. }
12. }

```

- a) 1 2 3 4 5 6 7 8 9 10
- b) 0 1 2 3 4 5 6 7 8 9 10
- c) i j k l m n o p q r
- d) i i i i i i i i i i

Answer: d

Explanation: None.

output:

```

$ javac array_output.java
$ java array_output
i i i i i i i i i i

```

10. What will be the output of the following Java code?

```

1.    class array_output
2.    {
3.        public static void main(String args[])
4.        {
5.            int array_variable[][] = {{ 1, 2, 3}, { 4 , 5, 6}, { 7, 8, 9}};
6.            int sum = 0;
7.            for (int i = 0; i < 3; ++i)
8.                for (int j = 0; j < 3 ; ++j)
9.                    sum = sum + array_variable[i][j];
10.           System.out.print(sum / 5);
11.       }
12.   }

```

- a) 8
- b) 9
- c) 10
- d) 11

Answer: b

Explanation: None.

output:

```

$ javac array_output.java
$ java array_output
9

```

1. Which of the following can be operands of arithmetic operators?

- a) Numeric

- b) Boolean
- c) Characters
- d) Both Numeric & Characters

Answer: d

Explanation: The operand of arithmetic operators can be any of numeric or character type, But not boolean.

2. Modulus operator, %, can be applied to which of these?

- a) Integers
- b) Floating – point numbers
- c) Both Integers and floating – point numbers
- d) None of the mentioned

Answer: c

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

3. With x = 0, which of the following are legal lines of Java code for changing the value of x to 1?

```
1. x++;  
2. x = x + 1;  
3. x += 1;  
4. x =+ 1;
```

- a) 1, 2 & 3
- b) 1 & 4
- c) 1, 2, 3 & 4
- d) 3 & 2

Answer: c

Explanation: Operator ++ increases value of variable by 1. $x = x + 1$ can also be written in shorthand form as $x += 1$. Also $x =+ 1$ will set the value of x to 1.

4. Decrement operator, —, decreases the value of variable by what number?

- a) 1
- b) 2
- c) 3
- d) 4

Answer: a

Explanation: None.

5. Which of these statements are incorrect?

- a) Assignment operators are more efficiently implemented by Java run-time system than their equivalent long forms
- b) Assignment operators run faster than their equivalent long forms
- c) Assignment operators can be used only with numeric and character data type
- d) None of the mentioned

Answer: d

Explanation: None.

6. What will be the output of the following Java program?

```
1.      class increment  
2.      {  
3.          public static void main(String args[])  
4.          {
```

```

5.         double var1 = 1 + 5;
6.         double var2 = var1 / 4;
7.         int var3 = 1 + 5;
8.         int var4 = var3 / 4;
9.         System.out.print(var2 + " " + var4);
10.
11.     }
12. }

```

- a) 1 1
- b) 0 1
- c) 1.5 1
- d) 1.5 1.0

Answer: c

Explanation: None

output:

```

$ javac increment.java
$ java increment
1.5 1

```

7. What will be the output of the following Java program?

```

1.     class Modulus
2.     {
3.         public static void main(String args[])
4.         {
5.             double a = 25.64;
6.             int b = 25;
7.             a = a % 10;
8.             b = b % 10;
9.             System.out.println(a + " " + b);
10.        }
11.    }

```

- a) 5.6400000000000001 5
- b) 5.6400000000000001 5.0
- c) 5 5
- d) 5 5.6400000000000001

Answer: a

Explanation: Modulus operator returns the remainder of a division operation on the operand. $a = a \% 10$ returns 25.64 % 10 i.e 5.6400000000000001. Similarly $b = b \% 10$ returns 5.

output:

```

$ javac Modulus.java

```

```
$ java Modulus
5.6400000000000001 5
```

8. What will be the output of the following Java program?

```
1.      class increment
2.      {
3.          public static void main(String args[])
4.          {
5.              int g = 3;
6.              System.out.print(++g * 8);
7.          }
8.      }
```

- a) 25
- b) 24
- c) 32
- d) 33

Answer: c

*Explanation: Operator ++ has more preference than *, thus g becomes 4 and when multiplied by 8 gives 32.*
output:

```
$ javac increment.java
$ java increment
32
```

Answer: a

Explanation: Both data types have different memory representation that's why 8-byte integral data type can be stored to 4-byte floating point data type.

10. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              int a = 1;
6.              int b = 2;
7.              int c;
8.              int d;
9.              c = ++b;
10.             d = a++;
11.             c++;
12.             b++;
13.             ++a;
```

```
14.         System.out.println(a + " " + b + " " + c);
```

```
15.         }
```

```
16.     }
```

- a) 3 2 4
- b) 3 2 3
- c) 2 3 4
- d) 3 4 4

Answer: d

Explanation: None.

output:

```
$ javac Output.java
$ java Output
3 4 4
```

1. Which of these is not a bitwise operator?

- a) &
- b) &=
- c) |=
- d) <=

Answer: d

Explanation: <= is a relational operator.

2. Which operator is used to invert all the digits in a binary representation of a number?

- a) ~
- b) <<<
- c) >>>
- d) ^

Answer: a

Explanation: Unary not operator, ~, inverts all of the bits of its operand in binary representation.

3. On applying Left shift operator, <<, on integer bits are lost one they are shifted past which position bit?

- a) 1
- b) 32
- c) 33
- d) 31

Answer: d

Explanation: The left shift operator shifts all of the bits in a value to the left specified number of times. For each shift left, the high order bit is shifted out and lost, zero is brought in from the right. When a left shift is applied to an integer operand, bits are lost once they are shifted past the bit position 31.

4. Which right shift operator preserves the sign of the value?

- a) <<
- b) >>
- c) <<=
- d) >>=

Answer: b

Explanation: None.

5. Which of these statements are incorrect?

- a) The left shift operator, <<, shifts all of the bits in a value to the left specified number of times

- b) The right shift operator, `>>`, shifts all of the bits in a value to the right specified number of times
- c) The left shift operator can be used as an alternative to multiplying by 2
- d) The right shift operator automatically fills the higher order bits with 0

Answer: d

Explanation: The right shift operator automatically fills the higher order bit with its previous contents each time a shift occurs. This also preserves the sign of the value.

6. What will be the output of the following Java program?

```
1.    class bitwise_operator
2.    {
3.        public static void main(String args[])
4.        {
5.            int var1 = 42;
6.            int var2 = ~var1;
7.            System.out.print(var1 + " " + var2);
8.        }
9.    }
```

- a) 42 42
- b) 43 43
- c) 42 -43
- d) 42 43

Answer: c

Explanation: Unary not operator, `~`, inverts all of the bits of its operand. 42 in binary is 00101010 in using `~` operator on var1 and assigning it to var2 we get inverted value of 42 i.e 11010101 which is -43 in decimal.

output:

```
$ javac bitwise_operator.java
$ java bitwise_operator
42 -43
```

7. What will be the output of the following Java program?

```
1.    class bitwise_operator
2.    {
3.        public static void main(String args[])
4.        {
5.            int a = 3;
6.            int b = 6;
7.            int c = a | b;
8.            int d = a & b;
9.            System.out.println(c + " " + d);
10.    }
```

```
11.      }
```

- a) 7 2
- b) 7 7
- c) 7 5
- d) 5 2

Answer: a

Explanation: And operator produces 1 bit if both operand are 1. Or operator produces 1 bit if any bit of the two operands in 1.

output:

```
$ javac bitwise_operator.java
$ java bitwise_operator
7 2
```

8. What will be the output of the following Java program?

```
1.      class leftshift_operator
2.      {
3.          public static void main(String args[])
4.          {
5.              byte x = 64;
6.              int i;
7.              byte y;
8.              i = x << 2;
9.              y = (byte) (x << 2)
10.             System.out.print(i + " " + y);
11.         }
12.     }
```

- a) 0 64
- b) 64 0
- c) 0 256
- d) 256 0

Answer: d

Explanation: None.

output:

```
$ javac leftshift_operator.java
$ java leftshift_operator
256 0
```

9. What will be the output of the following Java program?

```
1.      class rightshift_operator
2.      {
3.          public static void main(String args[])
```

```

4.      {
5.          int x;
6.          x = 10;
7.          x = x >> 1;
8.          System.out.println(x);
9.      }
10.     }

```

- a) 10
- b) 5
- c) 2
- d) 20

Answer: b

*Explanation: Right shift operator, >>, divides the value by 2.
output:*

```

$ javac rightshift_operator.java
$ java rightshift_operator
5

```

10. What will be the output of the following Java program?

```

1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              int a = 1;
6.              int b = 2;
7.              int c = 3;
8.              a |= 4;
9.              b >>= 1;
10.             c <<= 1;
11.             a ^= c;
12.             System.out.println(a + " " + b + " " + c);
13.         }
14.     }

```

- a) 3 1 6
- b) 2 2 3
- c) 2 3 4
- d) 3 3 6

Answer: a

Explanation: None.

output:

```
$ javac Output.java
$ java Output
3 1 6
```

1. What is the output of relational operators?

- a) Integer
- b) Boolean
- c) Characters
- d) Double

Answer: b

Explanation: None.

2. Which of these is returned by “greater than”, “less than” and “equal to” operators?

- a) Integers
- b) Floating – point numbers
- c) Boolean
- d) None of the mentioned

Answer: c

Explanation: All relational operators return a boolean value ie. true and false.

3. Which of the following operators can operate on a boolean variable?

- 1. &&
- 2. ==
- 3. ?:
- 4. +=

- a) 3 & 2
- b) 1 & 4
- c) 1, 2 & 4
- d) 1, 2 & 3

Answer: d

Explanation: Operator Short circuit AND, &&, equal to, ==, ternary if-then-else, ?:, are boolean logical operators. += is an arithmetic operator it can operate only on numeric values.

4. Which of these operators can skip evaluating right hand operand?

- a) !
- b) |
- c) &
- d) &&

Answer: d

Explanation: Operator short circuit and, &&, and short circuit or, ||, skip evaluating right hand operand when output can be determined by left operand alone.

5. Which of these statements is correct?

- a) true and false are numeric values 1 and 0
- b) true and false are numeric values 0 and 1
- c) true is any non zero value and false is 0
- d) true and false are non numeric values

Answer: d

Explanation: True and false are keywords, they are non numeric values which do not relate to zero or non zero numbers. true and false are boolean values.

6. What will be the output of the following Java code?

```
1.    class Relational_operator
2.    {
3.        public static void main(String args[])
4.        {
5.            int var1 = 5;
6.            int var2 = 6;
7.            System.out.print(var1 > var2);
8.        }
9.    }
```

- a) 1
- b) 0
- c) true
- d) false

Answer: d

Explanation: Operator > returns a boolean value. 5 is not greater than 6 therefore false is returned.
output:

```
$ javac Relational_operator.java
$ java Relational_operator
false
```

7. What will be the output of the following Java code?

```
1.    class bool_operator
2.    {
3.        public static void main(String args[])
4.        {
5.            boolean a = true;
6.            boolean b = !true;
7.            boolean c = a | b;
8.            boolean d = a & b;
9.            boolean e = d ? b : c;
10.           System.out.println(d + " " + e);
11.        }
12.    }
```

- a) false false
- b) true ture
- c) true false
- d) false true

Answer: d

Explanation: Operator | returns true if any one operand is true, thus 'c = true | false' is true. Operator & returns a true if both of the operand is true thus d is false. Ternary operator ?: assigns left of ':' if condition is true and right hand of ':' if condition is false. d is false thus e = d ? b : c, assigns c to e, e contains true.

output:

```
$ javac bool_operator.java
$ java bool_operator
false true
```

8. What will be the output of the following Java code?

```
1.      class ternary_operator
2.      {
3.          public static void main(String args[])
4.          {
5.              int x = 3;
6.              int y = ~ x;
7.              int z;
8.              z = x > y ? x : y;
9.              System.out.print(z);
10.         }
11.     }
```

- a) 0
- b) 1
- c) 3
- d) -4

Answer: c

Explanation: None.

output:

```
$ javac ternary_operator.java
$ java ternary_operator
3
```

9. What will be the output of the following Java code?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              int x , y = 1;
6.              x = 10;
7.              if (x != 10 && x / 0 == 0)
8.                  System.out.println(y);
```

```

9.         else
10.             System.out.println(++y);
11.     }
12. }

```

- a) 1
- b) 2
- c) Runtime error owing to division by zero in if condition
- d) Unpredictable behavior of program

Answer: b

Explanation: Operator short circuit and, &&, skips evaluating right hand operand if left hand operand is false thus division by zero in if condition does not give an error.

output:

```

$ javac Output.java
$ java Output
2

```

10. What will be the output of the following Java code?

```

1.     class Output
2.     {
3.         public static void main(String args[])
4.         {
5.             boolean a = true;
6.             boolean b = false;
7.             boolean c = a ^ b;
8.             System.out.println(!c);
9.         }
10.    }

```

- a) 0
- b) 1
- c) false
- d) true

Answer: c

Explanation: None.

output:

```

$ javac Output.java
$ java Output
false

```

1. Which of these have highest precedence?

- a) ()
- b) ++
- c) *
- d) >>

Answer: a

Explanation: Order of precedence is (highest to lowest) $a \rightarrow b \rightarrow c \rightarrow d$.

2. What should be expression1 evaluate to in using ternary operator as in this line?

```
expression1 ? expression2 : expression3
```

- a) Integer
- b) Floating – point numbers
- c) Boolean
- d) None of the mentioned

Answer: c

Explanation: The controlling condition of ternary operator must evaluate to boolean.

3. What is the value stored in x in the following lines of Java code?

```
int x, y, z;  
x = 0;  
y = 1;  
x = y = z = 8;
```

- a) 0
- b) 1
- c) 9
- d) 8

Answer: d

Explanation: None.

4. What is the order of precedence (highest to lowest) of following operators?

```
1. &  
2. ^  
3. ?:
```

- a) 1 -> 2 -> 3
- b) 2 -> 1 -> 3
- c) 3 -> 2 -> 1
- d) 2 -> 3 -> 1

Answer: a

Explanation: None.

5. Which of these statements are incorrect?

- a) Equal to operator has least precedence
- b) Brackets () have highest precedence
- c) Division operator, /, has higher precedence than multiplication operator
- d) Addition operator, +, and subtraction operator have equal precedence

Answer: c

Explanation: Division operator, /, has equal precedence as of multiplication operator. In expression involving multiplication and division evaluation of expression will begin from the right side when no brackets are used.

6. What will be the output of the following Java code?

```
1.      class operators  
2.      {  
3.          public static void main(String args[])
```

```

4.      {
5.          int var1 = 5;
6.          int var2 = 6;
7.          int var3;
8.          var3 = ++ var2 * var1 / var2 + var2;
9.          System.out.print(var3);
10.     }
11. }

```

- a) 10
- b) 11
- c) 12
- d) 56

Answer: c

Explanation: Operator ++ has the highest precedence than /, * and +. var2 is incremented to 7 and then used in expression, var3 = 7 * 5 / 7 + 7, gives 12.

output:

```

$ javac operators.java
$ java operators
12

```

7. What will be the output of the following Java code?

```

1.  class operators
2.  {
3.      public static void main(String args[])
4.      {
5.          int x = 8;
6.          System.out.println(++x * 3 + " " + x);
7.      }
8.  }

```

- a) 24 8
- b) 24 9
- c) 27 8
- d) 27 9

Answer: d

Explanation: Operator ++ has higher precedence than multiplication operator, *, x is incremented to 9 then multiplied with 3 giving 27.

output:

```

$ javac operators.java
$ java operators
27 9

```

8. What will be the output of the following Java code?

```

1. class Output
2. {
3.     public static void main(String args[])
4.     {
5.         int x=y=z=20;
6.
7.     }
8. }

```

- a) compile and runs fine
- b) 20
- c) run time error
- d) compile time error

Answer: d

Explanation: None.

9. Which of these lines of Java code will give better performance?

```

1. a | 4 + c >> b & 7;
2. (a | ((( 4 * c ) >> b ) & 7 ))

```

- a) 1 will give better performance as it has no parentheses
- b) 2 will give better performance as it has parentheses
- c) Both 1 & 2 will give equal performance
- d) Dependent on the computer system

Answer: c

Explanation: Parentheses do not degrade the performance of the program. Adding parentheses to reduce ambiguity does not negatively affect your system.

10. What will be the output of the following Java program?

```

1. class Output
2. {
3.     public static void main(String args[])
4.     {
5.         int a,b,c,d;
6.         a=b=c=d=20
7.         a+=b-=c*=d/=20
8.         System.out.println(a+" "+b+" "+c+" "+d);
9.
10.    }
11. }

```

- a) compile time error
- b) runtime error

- c) a=20 b=0 c=20 d=1
d) none of the mentioned

Answer: c

Explanation: Expression will evaluate from right to left.

output:

```
$ javac Output.java
$ java Output
20 0 20 1
```

1. Which of these selection statements test only for equality?

- a) if
b) switch
c) if & switch
d) none of the mentioned

Answer: b

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

2. Which of these are selection statements in Java?

- a) if()
b) for()
c) continue
d) break

Answer: a

Explanation: Continue and break are jump statements, and for is a looping statement.

3. Which of the following loops will execute the body of loop even when condition controlling the loop is initially false?

- a) do-while
b) while
c) for
d) none of the mentioned

Answer: a

Explanation: None.

4. Which of these jump statements can skip processing the remainder of the code in its body for a particular iteration?

- a) break
b) return
c) exit
d) continue

Answer: d

Explanation: None.

5. Which of this statement is incorrect?

- a) switch statement is more efficient than a set of nested ifs
b) two case constants in the same switch can have identical values
c) switch statement can only test for equality, whereas if statement can evaluate any type of boolean expression
d) it is possible to create a nested switch statements

Answer: b

Explanation: No two case constants in the same switch can have identical values.

6. What will be the output of the following Java program?

```

1.     class selection_statements
2.     {
3.         public static void main(String args[])
4.         {
5.             int var1 = 5;
6.             int var2 = 6;
7.             if ((var2 = 1) == var1)
8.                 System.out.print(var2);
9.             else
10.                 System.out.print(++var2);
11.         }
12.     }

```

- a) 1
- b) 2
- c) 3
- d) 4

Answer: b

*Explanation: var2 is initialised to 1. The conditional statement returns false and the else part gets executed.
output:*

```

$ javac selection_statements.java
$ java selection_statements
2

```

7. What will be the output of the following Java program?

```

1.     class comma_operator
2.     {
3.         public static void main(String args[])
4.         {
5.             int sum = 0;
6.             for (int i = 0, j = 0; i < 5 & j < 5; ++i, j = i + 1)
7.                 sum += i;
8.             System.out.println(sum);
9.         }
10.    }

```

- a) 5
- b) 6
- c) 14
- d) compilation error

Answer: b

Explanation: 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 $-0, 1, 2, 3, 4, 5$.
output:

```
$ javac comma_operator.java
$ java comma_operator
6
```

8. What will be the output of the following Java program?

```
1.      class jump_statments
2.      {
3.          public static void main(String args[])
4.          {
5.              int x = 2;
6.              int y = 0;
7.              for ( ; y < 10; ++y)
8.              {
9.                  if (y % x == 0)
10.                     continue;
11.                  else if (y == 8)
12.                     break;
13.                  else
14.                     System.out.print(y + " ");
15.              }
16.          }
17.      }
```

- a) 1 3 5 7
- b) 2 4 6 8
- c) 1 3 5 7 9
- d) 1 2 3 4 5 6 7 8 9

Answer: c

Explanation: 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.

output:

```
$ javac jump_statments.java
$ java jump_statments
1 3 5 7 9
```

9. What will be the output of the following Java program?

```
1. class Output
```



```

16.             System.out.println(a);
17.         }
18.             System.out.println(b);
19.         }
20.     }
21. }

```

- a) 5 10
- b) 10 5
- c) 5
- d) 10

Answer: d

Explanation: $b > a$ in if returns 5 which is equal to a i.e 5, therefore body of if is executed and block second is exited. Control goes to end of the block second executing the last print statement, printing 10.

output:

```

$ javac Output.java
$ java Output
10

```

1. What would be the output of the following code snippet if variable a=10?

```

1. if(a<=0)
2. {
3.     if(a==0)
4.     {
5.         System.out.println("1 ");
6.     }
7.     else
8.     {
9.         System.out.println("2 ");
10.    }
11. }
12. System.out.println("3 ");

```

- a) 1 2
- b) 2 3
- c) 1 3
- d) 3

Answer: d

Explanation: Since the first if condition is not met, control would not go inside if statement and hence only statement after the entire if block will be executed.

Answer: b

Explanation: While loop repeats a set of code only until the condition is met.

3. What is true about a break?

- a) Break stops the execution of entire program
- b) Break halts the execution and forces the control out of the loop
- c) Break forces the control out of the loop and starts the execution of next iteration
- d) Break halts the execution of the loop for certain time frame

Answer: b

Explanation: Break halts the execution and forces the control out of the loop.

4. What is true about do statement?

- a) do statement executes the code of a loop at least once
- b) do statement does not get execute if condition is not matched in the first iteration
- c) do statement checks the condition at the beginning of the loop
- d) do statement executes the code more than once always

Answer: a

Explanation: Do statement checks the condition at the end of the loop. Hence, code gets executed at least once.

5. Which of the following is used with the switch statement?

- a) Continue
- b) Exit
- c) break
- d) do

Answer: c

Explanation: Break is used with a switch statement to shift control out of switch.

6. What is the valid data type for variable “a” to print “Hello World”?

```
1. switch(a)
2. {
3.     System.out.println("Hello World");
4. }
```

- a) int and float
- b) byte and short
- c) char and long
- d) byte and char

Answer: d

Explanation: The switch condition would only meet if variable “a” is of type byte or char.

7. Which of the following is not a decision making statement?

- a) if
- b) if-else
- c) switch
- d) do-while

Answer: d

Explanation: do-while is an iteration statement. Others are decision making statements.

8. Which of the following is not a valid jump statement?

- a) break
- b) goto
- c) continue
- d) return

Answer: b

Explanation: break, continue and return transfer control to another part of the program and returns back to caller after execution. However, goto is marked as not used in Java.

9. From where break statement causes an exit?

- a) Only from innermost loop**
- b) Terminates a program**
- c) Only from innermost switch**
- d) From innermost loops or switches**

Answer: d

Explanation: The break statement causes an exit from innermost loop or switch.

10. Which of the following is not a valid flow control statement?

- a) exit()**
- b) break**
- c) continue**
- d) return**

Answer: a

Explanation: exit() is not a flow control statement in Java. exit() terminates the currently running JVM.

1. Which of the following is not OOPS concept in Java?

- a) Inheritance**
- b) Encapsulation**
- c) Polymorphism**
- d) Compilation**

Answer: d

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

2. Which of the following is a type of polymorphism in Java?

- a) Compile time polymorphism**
- b) Execution time polymorphism**
- c) Multiple polymorphism**
- d) Multilevel polymorphism**

Answer: a

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

3. When does method overloading is determined?

- a) At run time**
- b) At compile time**
- c) At coding time**
- d) At execution time**

Answer: b

Explanation: Overloading is determined at compile time. Hence, it is also known as compile time polymorphism.

4. When Overloading does not occur?

- a) More than one method with same name but different method signature and different number or type of parameters**
- b) More than one method with same name, same signature but different number of signature**
- c) More than one method with same name, same signature, same number of parameters but different type**
- d) More than one method with same name, same number of parameters and type but different signature**

Answer: d

Explanation: 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.

5. Which concept of Java is a way of converting real world objects in terms of class?

- a) Polymorphism**
- b) Encapsulation**
- c) Abstraction**
- d) Inheritance**

Answer: c

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

6. Which concept of Java is achieved by combining methods and attribute into a class?

- a) Encapsulation**
- b) Inheritance**
- c) Polymorphism**
- d) Abstraction**

Answer: a

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

7. What is it called if an object has its own lifecycle and there is no owner?

- a) Aggregation**
- b) Composition**
- c) Encapsulation**
- d) Association**

Answer: d

Explanation: 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.

8. What is it called where child object gets killed if parent object is killed?

- a) Aggregation**
- b) Composition**
- c) Encapsulation**
- d) Association**

Answer: b

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

9. What is it called where object has its own lifecycle and child object cannot belong to another parent object?

- a) Aggregation**
- b) Composition**
- c) Encapsulation**
- d) Association**

Answer: a

Explanation: Aggregation occurs when objects have their own life cycle and child object can associate with only one parent object.

Answer: a
Explanation: In order for method overriding, method with same signature in both superclass and subclass is required with same signature. That satisfies both concepts inheritance and polymorphism.

1. Which component is used to compile, debug and execute java program?

- a) JVM**
- b) JDK**
- c) JIT**
- d) JRE**

Answer: b

Explanation: JDK is a core component of Java Environment and provides all the tools, executables and binaries required to compile, debug and execute a Java Program.

2. Which component is responsible for converting bytecode into machine specific code?

- a) JVM**
- b) JDK**
- c) JIT**
- d) JRE**

Answer: a

Explanation: JVM is responsible to converting bytecode to the machine specific code. JVM is also platform dependent and provides core java functions like garbage collection, memory management, security etc.

3. Which component is responsible to run java program?

- a) JVM**
- b) JDK**
- c) JIT**
- d) JRE**

Answer: d

Explanation: JRE is the implementation of JVM, it provides platform to execute java programs.

4. Which component is responsible to optimize bytecode to machine code?

- a) JVM**
- b) JDK**
- c) JIT**
- d) JRE**

Answer: c

Explanation: 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.

5. Which statement is true about java?

- a) Platform independent programming language**
- b) Platform dependent programming language**
- c) Code dependent programming language**
- d) Sequence dependent programming language**

Answer: a

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

6. Which of the below is invalid identifier with the main method?

- a) public**
- b) static**
- c) private**
- d) final**

Answer: c

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

7. What is the extension of java code files?

- a) .class**
- b) .java**
- c) .txt**
- d) .js**

Answer: b

Explanation: Java files have .java extension.

8. What is the extension of compiled java classes?

- a) .class
- b) .java
- c) .txt
- d) .js

Answer: a

Explanation: The compiled java files have .class extension.

9. How can we identify whether a compilation unit is class or interface from a .class file?

- a) Java source file header
- b) Extension of compilation unit
- c) We cannot differentiate between class and interface
- d) The class or interface name should be postfixed with unit type

Answer: a

Explanation: The Java source file contains a header that declares the type of class or interface, its visibility with respect to other classes, its name and any superclass it may extend, or interface it implements.

10. What is use of interpreter?

- a) They convert bytecode to machine language code
- b) They read high level code and execute them
- c) They are intermediated between JIT and JVM
- d) It is a synonym for JIT

Answer: b

Explanation: Interpreters read high level language (interprets it) and execute the program. Interpreters are normally not passing through byte-code and jit compilation.

1. What is the stored in the object obj in following lines of Java code?

```
box obj;
```

- a) Memory address of allocated memory of object
- b) NULL
- c) Any arbitrary pointer
- d) Garbage

Answer: b

Explanation: Memory is allocated to an object using new operator. box obj; just declares a reference to object, no memory is allocated to it hence it points to NULL.

2. Which of these keywords is used to make a class?

- a) class
- b) struct
- c) int
- d) none of the mentioned

Answer: a

Explanation: None.

3. Which of the following is a valid declaration of an object of class Box?

- a) Box obj = new Box();
- b) Box obj = new Box;
- c) obj = new Box();

d) new Box obj;

Answer: a

Explanation: None.

4. Which of these operators is used to allocate memory for an object?

a) malloc

b) alloc

c) new

d) give

Answer: c

Explanation: Operator new dynamically allocates memory for an object and returns a reference to it. This reference is address in memory of the object allocated by new.

5. Which of these statement is incorrect?

a) Every class must contain a main() method

b) Applets do not require a main() method at all

c) There can be only one main() method in a program

d) main() method must be made public

Answer: a

Explanation: Every class does not need to have a main() method, there can be only one main() method which is made public.

6. What will be the output of the following Java program?

```
1.      class main_class
2.      {
3.          public static void main(String args[])
4.          {
5.              int x = 9;
6.              if (x == 9)
7.              {
8.                  int x = 8;
9.                  System.out.println(x);
10.             }
11.         }
12.     }
```

a) 9

b) 8

c) Compilation error

d) Runtime error

Answer: c

Explanation: Two variables with the same name can't be created in a class.

output:

```
$ javac main_class.java
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
```

7. Which of the following statements is correct?

- a) Public method is accessible to all other classes in the hierarchy**
- b) Public method is accessible only to subclasses of its parent class**
- c) Public method can only be called by object of its class**
- d) Public method can be accessed by calling object of the public class**

Answer: a

Explanation: None.

8. What will be the output of the following Java program?

```
1.      class box
2.      {
3.          int width;
4.          int height;
5.          int length;
6.      }
7.      class mainclass
8.      {
9.          public static void main(String args[])
10.         {
11.             box obj = new box();
12.             obj.width = 10;
13.             obj.height = 2;
14.             obj.length = 10;
15.             int y = obj.width * obj.height * obj.length;
16.             System.out.print(y);
17.         }
18.     }
```

- a) 12**
- b) 200**
- c) 400**
- d) 100**

Answer: b

Explanation: None.

output:

```
$ javac mainclass.java
$ java mainclass
200
```

9. What will be the output of the following Java program?

```
1.      class box
2.      {
3.          int width;
4.          int height;
5.          int length;
6.      }
7.      class mainclass
8.      {
9.          public static void main(String args[])
10.         {
11.             box obj1 = new box();
12.             box obj2 = new box();
13.             obj1.height = 1;
14.             obj1.length = 2;
15.             obj1.width = 1;
16.             obj2 = obj1;
17.             System.out.println(obj2.height);
18.         }
19.     }
```

- a) 1
- b) 2
- c) Runtime error
- d) Garbage value

Answer: a

Explanation: When we assign an object to another object of same type, all the elements of right side object gets copied to object on left side of equal to, =, operator.

output:

```
$ javac mainclass.java
$ java mainclass
1
```

10. What will be the output of the following Java program?

```
1.      class box
2.      {
3.          int width;
4.          int height;
5.          int length;
6.      }
```

```

7.      class mainclass
8.      {
9.          public static void main(String args[])
10.         {
11.             box obj = new box();
12.             System.out.println(obj);
13.         }
14.     }

```

- a) 0
- b) 1
- c) Runtime error
- d) [\[email protected\]](#) in hexadecimal form

Answer: d

Explanation: When we print object internally toString() will be called to return string into this format [\[email protected\]](#) in hexadecimal form.

output:

```

$ javac mainclass.java
$ java mainclass
\[email protected\]

```

1. What is the return type of a method that does not return any value?

- a) int
- b) float
- c) void
- d) double

Answer: c

Explanation: Return type of a method must be made void if it is not returning any value.

2. What is the process of defining more than one method in a class differentiated by method signature?

- a) Function overriding
- b) Function overloading
- c) Function doubling
- d) None of the mentioned

Answer: b

Explanation: Function overloading is a process of defining more than one method in a class with same name differentiated by function signature i.e return type or parameters type and number. Example – int volume(int length, int width) & int volume(int length, int width, int height) can be used to calculate volume.

3. Which of the following is a method having same name as that of it's class?

- a) finalize
- b) delete
- c) class
- d) constructor

Answer: d

Explanation: A constructor is a method that initializes an object immediately upon creation. It has the same name as that of class in which it resides.

4. Which method can be defined only once in a program?

- a) **main method**
- b) **finalize method**
- c) **static method**
- d) **private method**

Answer: a

Explanation: main() method can be defined only once in a program. Program execution begins from the main() method by java runtime system.

5. Which of this statement is incorrect?

- a) **All object of a class are allotted memory for the all the variables defined in the class**
- b) **If a function is defined public it can be accessed by object of other class by inheritance**
- c) **main() method must be made public**
- d) **All object of a class are allotted memory for the methods defined in the class**

Answer: d

Explanation: All object of class share a single copy of methods defined in a class, Methods are allotted memory only once. All the objects of the class have access to methods of that class are allotted memory only for the variables not for the methods.

6. What will be the output of the following Java program?

```
1.      class box
2.      {
3.          int width;
4.          int height;
5.          int length;
6.          int volume;
7.          void volume(int height, int length, int width)
8.          {
9.              volume = width*height*length;
10.         }
11.     }
12.     class Prameterized_method
13.     {
14.         public static void main(String args[])
15.         {
16.             box obj = new box();
17.             obj.height = 1;
18.             obj.length = 5;
19.             obj.width = 5;
20.             obj.volume(3,2,1);
21.             System.out.println(obj.volume);
```

22. }

23. }

- a) 0
- b) 1
- c) 6
- d) 25

Answer: c

Explanation: None.

output:

```
$ Prameterized_method.java
$ Prameterized_method
6
```

7. What will be the output of the following Java program?

```
1.      class equality
2.      {
3.          int x;
4.          int y;
5.          boolean isequal()
6.          {
7.              return(x == y);
8.          }
9.      }
10.     class Output
11.     {
12.         public static void main(String args[])
13.         {
14.             equality obj = new equality();
15.             obj.x = 5;
16.             obj.y = 5;
17.             System.out.println(obj.isequal());
18.         }
19.     }
```

- a) false
- b) true
- c) 0
- d) 1

Answer: b

Explanation: None.

output:

```
$ javac Output.java
$ java Output
true
```

8. What will be the output of the following Java program?

```
1.      class box
2.      {
3.          int width;
4.          int height;
5.          int length;
6.          int volume;
7.          void volume()
8.          {
9.              volume = width*height*length;
10.         }
11.     }
12.     class Output
13.     {
14.         public static void main(String args[])
15.         {
16.             box obj = new box();
17.             obj.height = 1;
18.             obj.length = 5;
19.             obj.width = 5;
20.             obj.volume();
21.             System.out.println(obj.volume);
22.         }
23.     }
```

- a) 0**
- b) 1**
- c) 25**
- d) 26**

Answer: c

Explanation: None.

output:

```
$ javac Output.java
$ java Output
```

9. In the following Java code, which call to sum() method is appropriate?

```

1. class Output
2. {
3.
4.     public static int sum(int ...x)
5.     {
6.         return;
7.     }
8.     static void main(String args[])
9.     {
10.         sum(10);
11.         sum(10,20);
12.         sum(10,20,30);
13.         sum(10,20,30,40);
14.     }
15. }
```

- a) only sum(10)**
- b) only sum(10,20)**
- c) only sum(10) & sum(10,20)**
- d) all of the mentioned**

Answer: d

Explanation: sum is a variable argument method and hence it can take any number as an argument.

10. What will be the output of the following Java program?

```

1.     class area
2.     {
3.         int width;
4.         int length;
5.         int volume;
6.         area()
7.         {
8.             width=5;
9.             length=6;
10.        }
11.        void volume()
```



```

12.         {
13.             volume = width*length*height;
14.         }
15.     }
16.     class cons_method
17.     {
18.         public static void main(String args[])
19.         {
20.             area obj = new area();
21.             obj.volume();
22.             System.out.println(obj.volume);
23.         }
24.     }

```

- a) 0
- b) 1
- c) 30
- d) error

Answer: d

Explanation: Variable height is not defined.

output:

```

$ javac cons_method.java
$ java cons_method
error: cannot find symbol height

```

1. What is the return type of Constructors?

- a) int
- b) float
- c) void
- d) none of the mentioned

Answer: d

Explanation: Constructors does not have any return type, not even void.

2. Which keyword is used by the method to refer to the object that invoked it?

- a) import
- b) catch
- c) abstract
- d) this

Answer: d

Explanation: this keyword can be used inside any method to refer to the current object. this is always a reference to the object on which the method was invoked.

3. Which of the following is a method having same name as that of its class?

- a) finalize
- b) delete
- c) class

d) constructor

Answer: d

Explanation: A constructor is a method that initializes an object immediately upon creation. It has the same name as that of class in which it resides.

4. Which operator is used by Java run time implementations to free the memory of an object when it is no longer needed?

- a) delete**
- b) free**
- c) new**
- d) none of the mentioned**

Answer: d

Explanation: Java handles deallocation of memory automatically, we do not need to explicitly delete an element. Garbage collection only occurs during execution of the program. When no references to the object exist, that object is assumed to be no longer needed, and the memory occupied by the object can be reclaimed.

5. Which function is used to perform some action when the object is to be destroyed?

- a) finalize()**
- b) delete()**
- c) main()**
- d) none of the mentioned**

Answer: a

Explanation: None.

6. What will be the output of the following Java code?

```
1.      class box
2.      {
3.          int width;
4.          int height;
5.          int length;
6.          int volume;
7.          box()
8.          {
9.              width = 5;
10.             height = 5;
11.             length = 6;
12.         }
13.         void volume()
14.         {
15.             volume = width*height*length;
16.         }
17.     }
```

```
18.     class constructor_output
19.     {
20.         public static void main(String args[])
21.         {
22.             box obj = new box();
23.             obj.volume();
24.             System.out.println(obj.volume);
25.         }
26.     }
```

- a) 100
- b) 150
- c) 200
- d) 250

Answer: b

Explanation: None.

output:

```
$ constructor_output.java
$ constructor_output
150
```

7. What will be the output of the following Java code?

```
1. class San
2. {
3.     San()throws IOException
4.     {
5.
6.     }
7.
8. }
9. class Foundry extends San
10. {
11.     Foundry()
12.     {
13.
14.     }
15.     public static void main(String[]args)
16.     {
```

```
17.  
18.     }  
19. }
```

- a) compile time error
- b) run time error
- c) compile and runs fine
- d) unreported exception java.io.IOException in default constructor

Answer: a

Explanation: If parent class constructor throws any checked exception, compulsory child class constructor should throw the same checked exception as its parent, otherwise code won't compile.

8. What will be the output of the following Java code?

```
1.     class box  
2.     {  
3.         int width;  
4.         int height;  
5.         int length;  
6.         int volume;  
7.         void finalize()  
8.         {  
9.             volume = width*height*length;  
10.            System.out.println(volume);  
11.        }  
12.        protected void volume()  
13.        {  
14.            volume = width*height*length;  
15.            System.out.println(volume);  
16.        }  
17.    }  
18.    class Output  
19.    {  
20.        public static void main(String args[])  
21.        {  
22.            box obj = new box();  
23.            obj.width=5;  
24.            obj.height=5;  
25.            obj.length=6;
```

```
26.         obj.volume();  
27.     }  
28. }
```

- a) 150
- b) 200
- c) Run time error
- d) Compilation error

Answer: a

Explanation: None.

output:

```
$ javac Output.java  
$ java Output  
150
```

9. Which of the following statements are incorrect?

- a) default constructor is called at the time of object declaration
- b) constructor can be parameterized
- c) finalize() method is called when a object goes out of scope and is no longer needed
- d) finalize() method must be declared protected

Answer: c

Explanation: finalize() method is called just prior to garbage collection. it is not called when object goes out of scope.

10. What will be the output of the following Java code?

```
1.     class area  
2.     {  
3.         int width;  
4.         int length;  
5.         int area;  
6.         void area(int width, int length)  
7.         {  
8.             this.width = width;  
9.             this.length = length;  
10.        }  
11.  
12.    }  
13.    class Output  
14.    {  
15.        public static void main(String args[])  
16.        {  
17.            area obj = new area();
```

```
18.         obj.area(5 , 6);

19.         System.out.println(obj.length + " " + obj.width);

20.     }

21. }
```

- a) 0 0
- b) 5 6
- c) 6 5
- d) 5 5

Answer: c

Explanation: this keyword can be used inside any method to refer to the current object. this is always a reference to the object on which the method was invoked.

output:

```
$ javac Output.java
$ java Output
6 5
```

1. What is true about private constructor?

- a) Private constructor ensures only one instance of a class exist at any point of time
- b) Private constructor ensures multiple instances of a class exist at any point of time
- c) Private constructor eases the instantiation of a class
- d) Private constructor allows creating objects in other classes

Answer: a

Explanation: Object of private constructor can only be created within class. Private constructor is used in singleton pattern.

2. What would be the behaviour if this() and super() used in a method?

- a) Runtime error
- b) Throws exception
- c) compile time error
- d) Runs successfully

Answer: c

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

3. What is false about constructor?

- a) Constructors cannot be synchronized in Java
- b) Java does not provide default copy constructor
- c) Constructor can have a return type
- d) “this” and “super” can be used in a constructor

Answer: c

Explanation: The constructor cannot have a return type. It should create and return new objects. Hence it would give a compilation error.

4. What is true about Class.getInstance()?

- a) Class.getInstance calls the constructor
- b) Class.getInstance is same as new operator
- c) Class.getInstance needs to have matching constructor
- d) Class.getInstance creates object if class does not have any constructor

Answer: d

Explanation: Class class provides list of methods for use like getInstance().

5. What is true about constructor?

- a) It can contain return type**
- b) It can take any number of parameters**
- c) It can have any non access modifiers**
- d) Constructor cannot throw an exception**

Answer: b

Explanation: Constructor returns a new object with variables defined as in the class. Instance variables are newly created and only one copy of static variables are created.

Answer: b

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

7. What is true about protected constructor?

- a) Protected constructor can be called directly**
- b) Protected constructor can only be called using super()**
- c) Protected constructor can be used outside package**
- d) protected constructor can be instantiated even if child is in a different package**

Answer: b

Explanation: Protected access modifier means that constructor can be accessed by child classes of the parent class and classes in the same package.

8. What is not the use of “this” keyword in Java?

- a) Passing itself to another method**
- b) Calling another constructor in constructor chaining**
- c) Referring to the instance variable when local variable has the same name**
- d) Passing itself to method of the same class**

Answer: d

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

9. What would be the behaviour if one parameterized constructor is explicitly defined?

- a) Compilation error**
- b) Compilation succeeds**
- c) Runtime error**
- d) Compilation succeeds but at the time of creating object using default constructor, it throws compilation error**

Answer: d

Explanation: The class compiles successfully. But the object creation of that class gives a compilation error.

10. What would be behaviour if the constructor has a return type?

- a) Compilation error**
- b) Runtime error**
- c) Compilation and runs successfully**
- d) Only String return type is allowed**

Answer: a

Explanation: The constructor cannot have a return type. It should create and return new object. Hence it would give compilation error.

1. Which of the following has the highest memory requirement?

- a) Heap**
- b) Stack**
- c) JVM**
- d) Class**

Answer: c

Explanation: JVM is the super set which contains heap, stack, objects, pointers, etc.

2. Where is a new object allocated memory?

- a) Young space
- b) Old space
- c) Young or Old space depending on space availability
- d) JVM

Answer: a

Explanation: A new object is always created in young space. Once young space is full, a special young collection is run where objects which have lived long enough are moved to old space and memory is freed up in young space for new objects.

3. Which of the following is a garbage collection technique?

- a) Cleanup model
- b) Mark and sweep model
- c) Space management model
- d) Sweep model

Answer: b

Explanation: A mark and sweep garbage collection consists of two phases, the mark phase and the sweep phase. In mark phase all the objects reachable by java threads, native handles and other root sources are marked alive and others are garbage. In sweep phase, the heap is traversed to find gaps between live objects and the gaps are marked free list used for allocating memory to new objects.

4. What is -Xms and -Xmx while starting jvm?

- a) Initial; Maximum memory
- b) Maximum; Initial memory
- c) Maximum memory
- d) Initial memory

Answer: a

Explanation: JVM will be started with Xms amount of memory and will be able to use a maximum of Xmx amount of memory. java -Xmx2048m -Xms256m.

5. Which exception is thrown when java is out of memory?

- a) MemoryFullException
- b) MemoryOutOfBoundsException
- c) OutOfMemoryError
- d) MemoryError

Answer: c

Explanation: The Xms flag has no default value, and Xmx typically has a default value of 256MB. A common use for these flags is when you encounter a java.lang.OutOfMemoryError.

6. How to get prints of shared object memory maps or heap memory maps for a given process?

- a) jmap
- b) memorymap
- c) memorypath
- d) jvmmmap

Answer: a

Explanation: We can use jmap as jmap -J-d64 -heap pid.

7. What happens to the thread when garbage collection kicks off?

- a) The thread continues its operation
- b) Garbage collection cannot happen until the thread is running
- c) The thread is paused while garbage collection runs
- d) The thread and garbage collection do not interfere with each other

Answer: c

Explanation: The thread is paused when garbage collection runs which slows the application performance.

8. Which of the below is not a Java Profiler?

- a) JVM**
- b) JConsole**
- c) JProfiler**
- d) Eclipse Profiler**

Answer: a

Explanation: Memory leak is like holding a strong reference to an object although it would never be needed anymore. Objects that are reachable but not live are considered memory leaks. Various tools help us to identify memory leaks.

9. Which of the below is not a memory leak solution?

- a) Code changes**
- b) JVM parameter tuning**
- c) Process restart**
- d) GC parameter tuning**

Answer: c

Explanation: Process restart is not a permanent fix to memory leak problem. The problem will resurge again.

Answer: b
Explanation: Garbage Collection cannot be controlled by a program.

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

- a) method overloading**
- b) method overriding**
- c) method hiding**
- d) none of the mentioned**

Answer: a

Explanation: 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.

2. Which of these can be overloaded?

- a) Methods**
- b) Constructors**
- c) All of the mentioned**
- d) None of the mentioned**

Answer: c

Explanation: None.

3. Which of these is correct about passing an argument by call-by-value process?

- a) Copy of argument is made into the formal parameter of the subroutine**
- b) Reference to original argument is passed to formal parameter of the subroutine**
- c) Copy of argument is made into the formal parameter of the subroutine and changes made on parameters of subroutine have effect on original argument**
- d) Reference to original argument is passed to formal parameter of the subroutine and changes made on parameters of subroutine have effect on original argument**

Answer: a

Explanation: 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.

4. What is the process of defining a method in terms of itself, that is a method that calls itself?

- a) Polymorphism**

- b) Abstraction
- c) Encapsulation
- d) Recursion

Answer: d

Explanation: None.

5. What will be the output of the following Java code?

```
1. class San
2. {
3.     public void m1 (int i,float f)
4.     {
5.         System.out.println(" int float method");
6.     }
7.
8.     public void m1(float f,int i);
9.     {
10.        System.out.println("float int method");
11.    }
12.
13.    public static void main(String[]args)
14.    {
15.        San s=new San();
16.        s.m1(20,20);
17.    }
18. }
```

- a) int float method
- b) float int method
- c) compile time error
- d) run time error

Answer: c

Explanation: While resolving overloaded method, compiler automatically promotes if exact match is not found. But in this case, which one to promote is an ambiguity.

6. What will be the output of the following Java code?

```
1.     class overload
2.     {
3.         int x;
4.         int y;
```

```

5.         void add(int a)
6.         {
7.             x = a + 1;
8.         }
9.         void add(int a, int b)
10.        {
11.            x = a + 2;
12.        }
13.    }
14.    class Overload_methods
15.    {
16.        public static void main(String args[])
17.        {
18.            overload obj = new overload();
19.            int a = 0;
20.            obj.add(6);
21.            System.out.println(obj.x);
22.        }
23.    }

```

- a) 5
- b) 6
- c) 7
- d) 8

Answer: c

Explanation: None.

output:

```

$ javac Overload_methods.java
$ java Overload_methods
7

```

7. What will be the output of the following Java code?

```

1.    class overload
2.    {
3.        int x;
4.        int y;
5.        void add(int a)
6.        {

```

```

7.         x = a + 1;
8.     }
9.     void add(int a , int b)
10.    {
11.        x = a + 2;
12.    }
13. }
14. class Overload_methods
15. {
16.     public static void main(String args[])
17.     {
18.         overload obj = new overload();
19.         int a = 0;
20.         obj.add(6, 7);
21.         System.out.println(obj.x);
22.     }
23. }

```

- a) 6
- b) 7
- c) 8
- d) 9

Answer: c

Explanation: None.

output:

```

$ javac Overload_methods.java
$ java Overload_methods
8

```

8. What will be the output of the following Java code?

```

1.     class overload
2.     {
3.         int x;
4.         double y;
5.         void add(int a , int b)
6.         {
7.             x = a + b;
8.         }

```

```

9.         void add(double c , double d)
10.        {
11.            y = c + d;
12.        }
13.        overload()
14.        {
15.            this.x = 0;
16.            this.y = 0;
17.        }
18.    }
19.    class Overload_methods
20.    {
21.        public static void main(String args[])
22.        {
23.            overload obj = new overload();
24.            int a = 2;
25.            double b = 3.2;
26.            obj.add(a, a);
27.            obj.add(b, b);
28.            System.out.println(obj.x + " " + obj.y);
29.        }
30.    }

```

- a) 6 6
- b) 6.4 6.4
- c) 6.4 6
- d) 4 6.4

Answer: d

Explanation: For obj.add(a,a); ,the function in line number 4 gets executed and value of x is 4. For the next function call, the function in line number 7 gets executed and value of y is 6.4

output:

```

$ javac Overload_methods.java
$ java Overload_methods
4 6.4

```

9. What will be the output of the following Java code?

```

1.    class test
2.    {
3.        int a;

```

```

4.         int b;
5.         void meth(int i , int j)
6.         {
7.             i *= 2;
8.             j /= 2;
9.         }
10.    }
11.    class Output
12.    {
13.        public static void main(String args[])
14.        {
15.            test obj = new test();
16.            int a = 10;
17.            int b = 20;
18.            obj.meth(a , b);
19.            System.out.println(a + " " + b);
20.        }
21.    }

```

- a) 10 20
- b) 20 10
- c) 20 40
- d) 40 20

Answer: a

Explanation: Variables a & b are passed by value, copy of their values are made on formal parameters of function meth() that is i & j. Therefore changes done on i & j are not reflected back on original arguments. a & b remain 10 & 20 respectively.

output:

```

$ javac Output.java
$ java Output
10 20

```

10. What will be the output of the following Java code?

```

1.    class test
2.    {
3.        int a;
4.        int b;
5.        test(int i, int j)
6.        {

```

```

7.         a = i;
8.         b = j;
9.     }
10.    void meth(test o)
11.    {
12.        o.a *= 2;
13.        o.b /= 2;
14.    }
15. }
16. class Output
17. {
18.     public static void main(String args[])
19.     {
20.         test obj = new test(10 , 20);
21.         obj.meth(obj);
22.         System.out.println(obj.a + " " + obj.b);
23.     }
24. }

```

- a) 10 20
- b) 20 10
- c) 20 40
- d) 40 20

Answer: b

Explanation: Class objects are always passed by reference, therefore changes done are reflected back on original arguments. obj.meth(obj) sends object obj as parameter whose variables a & b are multiplied and divided by 2 respectively by meth() function of class test. a & b becomes 20 & 10 respectively.

output:

```

$ javac Output.java
$ java Output
20 10

```

1. Which of these access specifiers must be used for main() method?

- a) private
- b) public
- c) protected
- d) none of the mentioned

Answer: b

Explanation: main() method must be specified public as it called by Java run time system, outside of the program. If no access specifier is used then by default member is public within its own package & cannot be accessed by Java run time system.

2. Which of these is used to access a member of class before object of that class is created?

- a) public
- b) private
- c) static
- d) protected

Answer: c

Explanation: None.

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

- a) private
- b) public
- c) public, within its own package
- d) protected

Answer: a

Explanation: 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.

4. What is the process by which we can control what parts of a program can access the members of a class?

- a) Polymorphism
- b) Abstraction
- c) Encapsulation
- d) Recursion

Answer: c

Explanation: None.

5. Which of the following statements are incorrect?

- a) public members of class can be accessed by any code in the program
- b) private members of class can only be accessed by other members of the class
- c) private members of class can be inherited by a subclass, and become protected members in subclass
- d) protected members of a class can be inherited by a subclass, and become private members of the subclass

Answer: c

Explanation: private members of a class can not be inherited by a subclass.

6. What will be the output of the following Java code?

```
1.      class access
2.      {
3.          public int x;
4.          private int y;
5.          void cal(int a, int b)
6.          {
7.              x =  a + 1;
8.              y =  b;
9.          }
10.     }
11.     public class access_specifier
12.     {
```



```

13.         public static void main(String args[])
14.         {
15.             access obj = new access();
16.             obj.cal(2, 3);
17.             System.out.println(obj.x + " " + obj.y);
18.         }
19.     }

```

- a) 3 3
- b) 2 3
- c) Runtime Error
- d) Compilation Error

Answer: c

Explanation: None.

output:

```

$ javac access_specifier.java
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    The field access.y is not visible

```

7. What will be the output of the following Java code?

```

1.     class access
2.     {
3.         public int x;
4.         private int y;
5.         void cal(int a, int b)
6.         {
7.             x =  a + 1;
8.             y =  b;
9.         }
10.        void print()
11.        {
12.            System.out.println(" " + y);
13.        }
14.    }
15.    public class access_specifier
16.    {
17.        public static void main(String args[])
18.        {

```

```
19.         access obj = new access();
20.
21.         obj.cal(2, 3);
22.
23.         System.out.println(obj.x);
24.
25.         obj.print();
26.
27.     }
28. }
```

- a) 2 3
- b) 3 3
- c) Runtime Error
- d) Compilation Error

Answer: b

Explanation: None.

output:

```
$ javac access_specifier.java
$ java access_specifier
3 3
```

8. What will be the output of the following Java code?

```
1.     class static_out
2.     {
3.
4.         static int x;
5.         static int y;
6.
7.         void add(int a, int b)
8.         {
9.
10.            x = a + b;
11.            y = x + b;
12.        }
13.    }
14.
15.    public class static_use
16.    {
17.
18.        public static void main(String args[])
19.        {
20.
21.            static_out obj1 = new static_out();
22.            static_out obj2 = new static_out();
23.
24.            int a = 2;
25.
26.            obj1.add(a, a + 1);
27.
28.            obj2.add(5, a);
```

```
20.          System.out.println(obj1.x + " " + obj2.y);
```

```
21.          }
```

```
22.      }
```

- a) 7 7.4
- b) 6 6.4
- c) 7 9
- d) 9 7

Answer: c

Explanation: None.

output:

```
$ javac static_use.java
$ java static_use
7 9
```

9. Which of these access specifier must be used for class so that it can be inherited by another subclass?

- a) public
- b) private
- c) protected
- d) none of the mentioned

Answer: a

Explanation: None.

1. Which one of the following is not an access modifier?

- a) Public
- b) Private
- c) Protected
- d) Void

Answer: d

Explanation: Public, private, protected and default are the access modifiers.

2. All the variables of class should be ideally declared as?

- a) private
- b) public
- c) protected
- d) default

Answer: a

Explanation: The variables should be private and should be accessed with get and set methods.

3. Which of the following modifier means a particular variable cannot be accessed within the package?

- a) private
- b) public
- c) protected
- d) default

Answer: a

Explanation: Private variables are accessible only within the class.

4. How can a protected modifier be accessed?

- a) accessible only within the class
- b) accessible only within package
- c) accessible within package and outside the package but through inheritance only

d) accessible by all

Answer: c

Explanation: The protected access modifier is accessible within package and outside the package but only through inheritance. The protected access modifier can be used with data member, method and constructor. It cannot be applied in the class.

5. What happens if constructor of class A is made private?

- a) Any class can instantiate objects of class A**
- b) Objects of class A can be instantiated only within the class where it is declared**
- c) Inherited class can instantiate objects of class A**
- d) classes within the same package as class A can instantiate objects of class A**

Answer: b

Explanation: If we make any class constructor private, we cannot create the instance of that class from outside the class.

6. All the variables of interface should be?

- a) default and final**
- b) default and static**
- c) public, static and final**
- d) protect, static and final**

Answer: c

Explanation: Variables of an interface are public, static and final by default because the interfaces cannot be instantiated, final ensures the value assigned cannot be changed with the implementing class and public for it to be accessible by all the implementing classes.

7. What is true of final class?

- a) Final class cause compilation failure**
- b) Final class cannot be instantiated**
- c) Final class cause runtime failure**
- d) Final class cannot be inherited**

Answer: d

Explanation: Final class cannot be inherited. This helps when we do not want classes to provide extension to these classes.

8. How many copies of static and class variables are created when 10 objects are created of a class?

- a) 1, 10**
- b) 10, 10**
- c) 10, 1**
- d) 1, 1**

Answer: a

Explanation: Only one copy of static variables are created when a class is loaded. Each object instantiated has its own copy of instance variables.

Answer: b
Explanation: Protected class member (method or variable) is like package-private (default visibility), except that it also can be accessed from subclasses. Since there is no such concept as 'subpackage' or 'package-inheritance' in Java, declaring class protected or package-private would be the same thing.

10. Which is the modifier when there is none mentioned explicitly?

- a) protected**
- b) private**
- c) public**
- d) default**

Answer: d

Explanation: Default is the access modifier when none is defined explicitly. It means the member (method or variable)

can be accessed within the same package.

1. Arrays in Java are implemented as?

- a) class**
- b) object**
- c) variable**
- d) none of the mentioned**

Answer: b

Explanation: None.

2. Which of these keywords is used to prevent content of a variable from being modified?

- a) final**
- b) last**
- c) constant**
- d) static**

Answer: a

Explanation: A variable can be declared final, doing so prevents its content from being modified. Final variables must be initialized when it is declared.

3. Which of these cannot be declared static?

- a) class**
- b) object**
- c) variable**
- d) method**

Answer: b

Explanation: static statements are run as soon as class containing them is loaded, prior to any object declaration.

4. Which of the following statements are incorrect?

- a) static methods can call other static methods only**
- b) static methods must only access static data**
- c) static methods can not refer to this or super in any way**
- d) when object of class is declared, each object contains its own copy of static variables**

Answer: d

Explanation: All objects of class share same static variable, when object of a class are declared, all the objects share same copy of static members, no copy of static variables are made.

5. Which of the following statements are incorrect?

- a) Variables declared as final occupy memory**
- b) final variable must be initialized at the time of declaration**
- c) Arrays in java are implemented as an object**
- d) All arrays contain an attribute-length which contains the number of elements stored in the array**

Answer: a

Explanation: None.

6. Which of these methods must be made static?

- a) main()**
- b) delete()**
- c) run()**
- d) finalize()**

Answer: a

Explanation: main() method must be declared static, main() method is called by Java runtime system before any object of any class exists.

7. What will be the output of the following Java program?

```
1.    class access
2.    {
3.        public int x;
4.        static int y;
5.        void cal(int a, int b)
6.        {
7.            x += a ;
8.            y += b;
9.        }
10.   }
11.   class static_specifier
12.   {
13.       public static void main(String args[])
14.       {
15.           access obj1 = new access();
16.           access obj2 = new access();
17.           obj1.x = 0;
18.           obj1.y = 0;
19.           obj1.cal(1, 2);
20.           obj2.x = 0;
21.           obj2.cal(2, 3);
22.           System.out.println(obj1.x + " " + obj2.y);
23.       }
24.   }
```

- a) 1 2
- b) 2 3
- c) 3 2
- d) 1 5

Answer: d

Explanation: None.

output:

```
$ javac static_specifier.java
$ java static_specifier
1 5
```

8. What will be the output of the following Java program?

```

1.     class access
2.     {
3.         static int x;
4.         void increment()
5.         {
6.             x++;
7.         }
8.     }
9.     class static_use
10.    {
11.        public static void main(String args[])
12.        {
13.            access obj1 = new access();
14.            access obj2 = new access();
15.            obj1.x = 0;
16.            obj1.increment();
17.            obj2.increment();
18.            System.out.println(obj1.x + " " + obj2.x);
19.        }
20.    }

```

- a) 1 2
- b) 1 1
- c) 2 2
- d) Compilation Error

Answer: c

Explanation: All objects of class share same static variable, all the objects share same copy of static members, obj1.x and obj2.x refer to same element of class which has been incremented twice and its value is 2.

output:

```

$ javac static_use.java
$ java static_use
2 2

```

9. What will be the output of the following Java program?

```

1.     class static_out
2.     {
3.         static int x;
4.         static int y;
5.         void add(int a , int b)

```

```

6.      {
7.          x = a + b;
8.          y = x + b;
9.      }
10. }
11. class static_use
12. {
13.     public static void main(String args[])
14.     {
15.         static_out obj1 = new static_out();
16.         static_out obj2 = new static_out();
17.         int a = 2;
18.         obj1.add(a, a + 1);
19.         obj2.add(5, a);
20.         System.out.println(obj1.x + " " + obj2.y);
21.     }
22. }

```

- a) 7 7
- b) 6 6
- c) 7 9
- d) 9 7

Answer: c

Explanation: None.

output:

```

$ javac static_use.java
$ java static_use
7 9

```

10. What will be the output of the following Java program?

```

1.  class Output
2.  {
3.      public static void main(String args[])
4.      {
5.          int arr[] = {1, 2, 3, 4, 5};
6.          for ( int i = 0; i < arr.length - 2; ++i)
7.              System.out.println(arr[i] + " ");
8.      }

```



```
9.      }
```

- a) 1 2
- b) 1 2 3
- c) 1 2 3 4
- d) 1 2 3 4 5

Answer: b

Explanation: `arr.length()` is 5, so the loop is executed for three times.

output:

```
$ javac Output.java
$ java Output
1 2 3
```

11. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              int a1[] = new int[10];
6.              int a2[] = {1, 2, 3, 4, 5};
7.              System.out.println(a1.length + " " + a2.length);
8.          }
9.      }
```

- a) 10 5
- b) 5 10
- c) 0 10
- d) 0 5

Answer: a

Explanation: Arrays in java are implemented as objects, they contain an attribute that is `length` which contains the number of elements that can be stored in the array. Hence `a1.length` gives 10 and `a2.length` gives 5.

output:

```
$ javac Output.java
$ java Output
10 5
```

1. String in Java is a?

- a) class
- b) object
- c) variable
- d) character array

Answer: a

Explanation: None.

2. Which of these method of String class is used to obtain character at specified index?

- a) `char()`
- b) `Charat()`
- c) `charat()`

d) charAt()

Answer: d

Explanation: None.

3. Which of these keywords is used to refer to member of base class from a subclass?

- a) upper**
- b) super**
- c) this**
- d) none of the mentioned**

Answer: b

Explanation: Whenever a subclass needs to refer to its immediate superclass, it can do so by use of the keyword super.

4. Which of these method of String class can be used to test to strings for equality?

- a) isequal()**
- b) isequals()**
- c) equal()**
- d) equals()**

Answer: d

Explanation: None.

5. Which of the following statements are incorrect?

- a) String is a class**
- b) Strings in java are mutable**
- c) Every string is an object of class String**
- d) Java defines a peer class of String, called StringBuffer, which allows string to be altered**

Answer: b

Explanation: Strings in Java are immutable that is they can not be modified.

6. What will be the output of the following Java program?

```
1.      class string_demo
2.      {
3.          public static void main(String args[])
4.          {
5.              String obj = "I" + "like" + "Java";
6.              System.out.println(obj);
7.          }
8.      }
```

- a) I**
- b) like**
- c) Java**
- d) IlikeJava**

Answer: d

Explanation: Java defines an operator +, it is used to concatenate strings.

output:

```
$ javac string_demo.java
$ java string_demo
```

7. What will be the output of the following Java program?

```
1.    class string_class
2.    {
3.        public static void main(String args[])
4.        {
5.            String obj = "I LIKE JAVA";
6.            System.out.println(obj.charAt(3));
7.        }
8.    }
```

- a) I
- b) L
- c) K
- d) E

Answer: a

Explanation: charAt() is a method of class String which gives the character specified by the index. obj.charAt(3) gives 4th character i.e I.

output:

```
$ javac string_class.java
$ java string_class
I
```

8. What will be the output of the following Java program?

```
1.    class string_class
2.    {
3.        public static void main(String args[])
4.        {
5.            String obj = "I LIKE JAVA";
6.            System.out.println(obj.length());
7.        }
8.    }
```

- a) 9
- b) 10
- c) 11
- d) 12

Answer: c

Explanation: None.

output:

```
$ javac string_class.java
$ java string_class
11
```

9. What will be the output of the following Java program?

```
1.    class string_class
2.    {
3.        public static void main(String args[])
4.        {
5.            String obj = "hello";
6.            String obj1 = "world";
7.            String obj2 = obj;
8.            obj2 = " world";
9.            System.out.println(obj + " " + obj2);
10.    }
11. }
```

- a) hello hello
- b) world world
- c) hello world
- d) world hello

Answer: c

Explanation: None.

output:

```
$ javac string_class.java
$ java string_class
hello world
```

10. What will be the output of the following Java program?

```
1.    class string_class
2.    {
3.        public static void main(String args[])
4.        {
5.            String obj = "hello";
6.            String obj1 = "world";
7.            String obj2 = "hello";
8.            System.out.println(obj.equals(obj1) + " " + obj.equals(obj2));
9.    }
10. }
```

- a) false false
- b) true true
- c) true false
- d) false true

Answer: d

Explanation: equals() is method of class String, it is used to check equality of two String objects, if they are equal, true is returned else false.

output:

```
$ javac string_class.java
$ java string_class
false true
```

1. Which of these is the method which is executed first before execution of any other thing takes place in a program?

- a) main method**
- b) finalize method**
- c) static method**
- d) private method**

Answer: c

Explanation: If a static method is present in the program then it will be executed first, then main will be executed.

2. What is the process of defining more than one method in a class differentiated by parameters?

- a) Function overriding**
- b) Function overloading**
- c) Function doubling**
- d) None of the mentioned**

Answer: b

Explanation: Function overloading is a process of defining more than one method in a class with same name differentiated by function signature i.e return type or parameters type and number. Example – int volume(int length, int width) & int volume(int length, int width, int height) can be used to calculate volume.

3. Which of these can be used to differentiate two or more methods having the same name?

- a) Parameters data type**
- b) Number of parameters**
- c) Return type of method**
- d) All of the mentioned**

Answer: d

Explanation: None.

4. Which of these data type can be used for a method having a return statement in it?

- a) void**
- b) int**
- c) float**
- d) both int and float**

Answer: d

Explanation: None.

5. Which of these statement is incorrect?

- a) Two or more methods with same name can be differentiated on the basis of their parameters data type**
- b) Two or more method having same name can be differentiated on basis of number of parameters**
- c) Any already defined method in java library can be defined again in the program with different data type of parameters**
- d) If a method is returning a value the calling statement must have a variable to store that value**

Answer: d

Explanation: Even if a method is returning a value, it is not necessary to store that value.

6. What will be the output of the following Java program?

```

1.     class box
2.     {
3.         int width;
4.         int height;
5.         int length;
6.         int volume;
7.         void volume(int height, int length, int width)
8.         {
9.             volume = width * height * length;
10.        }
11.    }
12.    class Prameterized_method{
13.        public static void main(String args[])
14.        {
15.            box obj = new box();
16.            obj.height = 1;
17.            obj.length = 5;
18.            obj.width = 5;
19.            obj.volume(3, 2, 1);
20.            System.out.println(obj.volume);
21.        }
22.    }

```

- a) 0
- b) 1
- c) 6
- d) 25

Answer: c

Explanation: None

output:

```

$ Prameterized_method.java
$ Prameterized_method
6

```

7. What will be the output of the following Java program?

```

1.     class equality
2.     {
3.         int x;

```

```

4.         int y;
5.         boolean isequal()
6.         {
7.             return(x == y);
8.         }
9.     }
10.    class Output
11.    {
12.        public static void main(String args[])
13.        {
14.            equality obj = new equality();
15.            obj.x = 5;
16.            obj.y = 5;
17.            System.out.println(obj.isequal);
18.        }
19.    }

```

- a) false
- b) true
- c) 0
- d) 1

Answer: b

Explanation: None

output:

```

$ javac Output.java
$ java Output
true

```

8. What will be the output of the following Java program?

```

1.    class box
2.    {
3.        int width;
4.        int height;
5.        int length;
6.        int volume;
7.        void volume()
8.        {
9.            volume = width * height * length;

```

```

10.         }
11.         void volume(int x)
12.         {
13.             volume = x;
14.         }
15.     }
16.     class Output
17.     {
18.         public static void main(String args[])
19.         {
20.             box obj = new box();
21.             obj.height = 1;
22.             obj.length = 5;
23.             obj.width = 5;
24.             obj.volume(5);
25.             System.out.println(obj.volume);
26.         }
27.     }

```

- a) 0
- b) 5
- c) 25
- d) 26

Answer: b

Explanation: None.

output:

```

$ javac Output.java
$ java Output
5

```

9. What will be the output of the following Java program?

```

1.     class Output
2.     {
3.         static void main(String args[])
4.         {
5.             int x , y = 1;
6.             x = 10;
7.             if(x != 10 && x / 0 == 0)

```



```
8.         System.out.println(y);
9.         else
10.         System.out.println(++y);
11.     }
12. }
```

a) 1

b) 2

c) Runtime Error

d) Compilation Error

Answer: d

Explanation: main() method must be made public. Without main() being public java run time system will not be able to access main() and will not be able to execute the code.

output:

```
$ javac Output.java
Error: Main method not found in class Output, please define the main method as:
    public static void main(String[] args)
```

10. What will be the output of the following Java program?

```
1.  class area
2.  {
3.      int width;
4.      int length;
5.      int height;
6.      area()
7.      {
8.          width = 5;
9.          length = 6;
10.         height = 1;
11.     }
12.     void volume()
13.     {
14.         volume = width * height * length;
15.     }
16. }
17. class cons_method
18. {
19.     public static void main(String args[])
20.     {
```

```

21.         area obj = new area();

22.         obj.volume();

23.         System.out.println(obj.volume);

24.     }

25. }

```

- a) 0
- b) 1
- c) 25
- d) 30

Answer: d

Explanation: None.

output:

```

$ javac cons_method.java
$ java cons_method
30

```

1. Which of this method is given parameter via command line arguments?

- a) main()
- b) recursive() method
- c) Any method
- d) System defined methods

Answer: a

Explanation: Only main() method can be given parameters via using command line arguments.

2. Which of these data types is used to store command line arguments?

- a) Array
- b) Stack
- c) String
- d) Integer

Answer: c

Explanation: None.

3. How many arguments can be passed to main()?

- a) Infinite
- b) Only 1
- c) System Dependent
- d) None of the mentioned

Answer: a

Explanation: None.

4. Which of these is a correct statement about args in the following line of code?

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

- a) args is a String
- b) args is a Character
- c) args is an array of String
- d) args in an array of Character

Answer: c

Explanation: args in an array of String.

5. Can command line arguments be converted into int automatically if required?

- a) Yes
- b) No
- c) Compiler Dependent
- d) Only ASCII characters can be converted

Answer: b

Explanation: All command Line arguments are passed as a string. We must convert numerical value to their internal forms manually.

6. What will be the output of the following Java program, Command line execution is done as – “java Output This is a command Line”?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            System.out.print("args[0]");
6.        }
7.    }
```

- a) java
- b) Output
- c) This
- d) is

Answer: c

Explanation: None.

Output:

```
$ javac Output.java
java Output This is a command Line
This
```

7. What will be the output of the following Java program, Command line execution is done as – “java Output This is a command Line”?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            System.out.print("args[3]");
6.        }
7.    }
```

- a) java
- b) is
- c) This

d) command

Answer: d

Explanation: None.

Output:

```
$ javac Output.javac
java Output This is a command Line
command
```

8. What will be the output of the following Java program, Command line execution is done as – “java Output This is a command Line”?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              System.out.print("args");
6.          }
7.      }
```

- a) This**
- b) java Output This is a command Line**
- c) This is a command Line**
- d) Compilation Error**

Answer: c

Explanation: None.

Output:

```
$ javac Output.javac
java Output This is a command Line
This is a command Line
```

9. What will be the output of the following Java program, Command line execution is done as – “java Output command Line 10 A b 4 N”?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              System.out.print("(int)args[2] * 2");
6.          }
7.      }
```

- a) java**
- b) 10**
- c) 20**
- d) b**

Answer: c

Explanation: None.

Output:

```
$ javac Output.javac
java Output command Line 10 A b 4 N
20
```

10. What will be the output of the following Java program, Command line execution is done as – “java Output command Line 10 A b 4 N”?

```
1. class Output
2. {
3.     public static void main(String args[])
4.     {
5.         System.out.print("args[6]");
6.     }
7. }
```

- a) java
- b) 10
- c) b
- d) N

Answer: d
Explanation: None.
Output:

```
$ javac Output.javac
java Output command Line 10 A b 4 N
N
```

1. What will be the output of the following Java snippet, if attempted to compile and run this code with command line argument “java abc Rakesh Sharma”?

```
1. public class abc
2. {
3.     int a=2000;
4.     public static void main(String argv[])
5.     {
6.         System.out.println(argv[1]+" :-Please pay Rs."+a);
7.     }
8. }
```

- a) Compile time error
- b) Compilation but runtime error
- c) Compilation and output Rakesh :-Please pay Rs.2000
- d) Compilation and output Sharma :-Please pay Rs.2000

Answer: a
Explanation: Main method is static and cannot access non static variable a.

2. What will be the output of the following Java snippet, if attempted to compile and execute?

```
1. class abc
2. {
3.     public static void main(String args[])
4.     {
5.         if(args.length>0)
6.             System.out.println(args.length);
7.     }
8. }
```

- a) The snippet compiles, runs and prints 0
- b) The snippet compiles, runs and prints 1
- c) The snippet does not compile
- d) The snippet compiles and runs but does not print anything

Answer: d

Explanation: As no argument is passed to the code, the length of args is 0. So the code will not print.

3. What will be the output of the following Java snippet, if compiled and executed with command line argument “java abc 1 2 3”?

```
1. public class abc
2. {
3.     static public void main(String [] xyz)
4.     {
5.         for(int n=1;n<xyz.length; n++)
6.         {
7.             System.out.println(xyz[n]+"");
8.         }
9.     }
10. }
```

- a) 1 2
- b) 2 3
- c) 1 2 3
- d) Compilation error

Answer: b

Explanation: The index of array starts with 0. Since the loop is starting with 1 it will print 2 3.

4. What will be the output of the following Java code snippet running with “java demo I write java code”?

```
1. public class demo
2. {
3.     public static void main(String args[])
```

```

4.      {
5.          System.out.println(args[0]+" "+args[args.length-1]);
6.      }
7.  }

```

- a) The snippet compiles, runs and prints “java demo”
- b) The snippet compiles, runs and prints “java code”
- c) The snippet compiles, runs and prints “demo code”
- d) The snippet compiles, runs and prints “I code”

Answer: d

Explanation: The index of array starts with 0 till length – 1. Hence it would print “I code”.

5. What will be the output of the following Java snippet, if compiled and executed with command line “hello there”?

```

1. public class abc
2. {
3.     String[] xyz;
4.
5.     public static void main(String argv[])
6.     {
7.         xyz=argv;
8.     }
9.
10.    public void runMethod()
11.    {
12.        System.out.println(argv[1]);
13.    }
14. }

```

- a) Compile time error
- b) Output would be “hello”
- c) Output would be “there”
- d) Output would be “hello there”

Answer: a

Explanation: Error would be “Cannot make static reference to a non static variable”. Even if main method was not static, the array argv is local to the main method and would not be visible within runMethod.

6. How do we pass command line argument in Eclipse?

- a) Arguments tab
- b) Variable tab
- c) Cannot pass command line argument in eclipse
- d) Environment variable tab

Answer: a

Explanation: Arguments tab is used to pass command line argument in eclipse.

7. Which class allows parsing of command line arguments?

- a) Args
- b) JCommander
- c) Command Line
- d) Input

Answer: b

Explanation: JCommander is a very small Java framework that makes it trivial to parse command line parameters.

8. Which annotation is used to represent command line input and assigned to correct data type?

- a) @Input
- b) @Variable
- c) @Command Line
- d) @Parameter

Answer: d

Explanation: @Parameter, @Parameter(names = { "-log", "-verbose" }, description = "Level of verbosity"), etc are various forms of using @Parameter

9. What will be the output of the following Java code snippet run as \$ java Demo --length 512 --breadth 2 -h 3?

```
1. class Demo {
2.     @Parameter(names={"--length"})
3.     int length;
4.
5.     @Parameter(names={"--breadth"})
6.     int breadth;
7.
8.     @Parameter(names={"--height", "-h"})
9.     int height;
10.
11.     public static void main(String args[])
12.     {
13.         Demo demo = new Demo();
14.         new JCommander(demo, args);
15.         demo.run();
16.     }
17.
18.     public void run()
19.     {
20.         System.out.println(length+" "+ breadth+" "+height);
21.     }
22. }
```


- a) 2 512 3
- b) 2 2 3
- c) 512 2 3
- d) 512 512 3

Answer: c

Explanation: JCommander helps easily pass command line arguments. @Parameter assigns input to desired parameter.

10. What is the use of @syntax?

- a) Allows multiple parameters to be passed
- b) Allows one to put all your options into a file and pass this file as a parameter
- c) Allows one to pass only one parameter
- d) Allows one to pass one file containing only one parameter

Answer: b

Explanation: JCommander supports the @syntax, which allows us to put all our options into a file and pass this file as a parameter.

```
/tmp/parameters
-verbose
file1
file2
$ java Main @/tmp/parameters
```

1. What is Recursion in Java?

- a) Recursion is a class
- b) Recursion is a process of defining a method that calls other methods repeatedly
- c) Recursion is a process of defining a method that calls itself repeatedly
- d) Recursion is a process of defining a method that calls other methods which in turn call again this method

Answer: b

Explanation: Recursion is the process of defining something in terms of itself. It allows us to define a method that calls itself.

2. Which of these data types is used by operating system to manage the Recursion in Java?

- a) Array
- b) Stack
- c) Queue
- d) Tree

Answer: b

Explanation: Recursions are always managed by using stack.

3. Which of these will happen if recursive method does not have a base case?

- a) An infinite loop occurs
- b) System stops the program after some time
- c) After 1000000 calls it will be automatically stopped
- d) None of the mentioned

Answer: a

Explanation: If a recursive method does not have a base case then an infinite loop occurs which results in Stack Overflow.

4. Which of these is not a correct statement?

- a) A recursive method must have a base case
- b) Recursion always uses stack
- c) Recursive methods are faster than programmers written loop to call the function repeatedly using a stack
- d) Recursion is managed by Java Runtime environment

Answer: d

Explanation: Recursion is always managed by operating system.

5. Which of these packages contains the exception Stack Overflow in Java?

- a) java.lang**
- b) java.util**
- c) java.io**
- d) java.system**

Answer: a

Explanation: None.

6. What will be the output of the following Java program?

```
1.      class recursion
2.      {
3.          int func (int n)
4.          {
5.              int result;
6.              result = func (n - 1);
7.              return result;
8.          }
9.      }
10.     class Output
11.     {
12.         public static void main(String args[])
13.         {
14.             recursion obj = new recursion() ;
15.             System.out.print(obj.func(12));
16.         }
17.     }
```

- a) 0**
- b) 1**
- c) Compilation Error**
- d) Runtime Error**

Answer: d

Explanation: Since the base case of the recursive function func() is not defined hence infinite loop occurs and results in Stack Overflow.

Output:

```
$ javac Output.java
$ java Output
Exception in thread "main" java.lang.StackOverflowError
```

7. What will be the output of the following Java program?

```

1.      class recursion
2.      {
3.          int func (int n)
4.          {
5.              int result;
6.              if (n == 1)
7.                  return 1;
8.              result = func (n - 1);
9.              return result;
10.         }
11.     }
12.     class Output
13.     {
14.         public static void main(String args[])
15.         {
16.             recursion obj = new recursion() ;
17.             System.out.print(obj.func(5));
18.         }
19.     }

```

- a) 0
- b) 1
- c) 120
- d) None of the mentioned

Answer: b

Explanation: None.

Output:

```

$ javac Output.java
$ java Output
1

```

8. What will be the output of the following Java program?

```

1.      class recursion
2.      {
3.          int fact(int n)
4.          {
5.              int result;
6.              if (n == 1)

```

```

7.         return 1;
8.         result = fact(n - 1) * n;
9.         return result;
10.    }
11. }
12. class Output
13. {
14.     public static void main(String args[])
15.     {
16.         recursion obj = new recursion() ;
17.         System.out.print(obj.fact(5));
18.     }
19. }

```

- a) 24
- b) 30
- c) 120
- d) 720

Answer: c

Explanation: *fact()* method recursively calculates factorial of a number, when value of *n* reaches 1, base case is executed and 1 is returned.

Output:

```

$ javac Output.javac
$ java Output
120

```

9. What will be the output of the following Java program?

```

1.     class recursion
2.     {
3.         int fact(int n)
4.         {
5.             int result;
6.             if (n == 1)
7.                 return 1;
8.             result = fact(n - 1) * n;
9.             return result;
10.        }
11.    }
12. class Output

```

```

13.      {
14.          public static void main(String args[])
15.      {
16.          recursion obj = new recursion() ;
17.          System.out.print(obj.fact(1));
18.      }
19.  }

```

- a) 1
- b) 30
- c) 120
- d) Runtime Error

Answer: a

Explanation: fact() method recursively calculates factorial of a number, when value of n reaches 1, base case is excuted and 1 is returned.

Output:

```

$ javac Output.java
$ java Output
1

```

10. What will be the output of the following Java program?

```

1.      class recursion
2.      {
3.          int fact(int n)
4.          {
5.              int result;
6.              if (n == 1)
7.                  return 1;
8.              result = fact(n - 1) * n;
9.              return result;
10.         }
11.     }
12.     class Output
13.     {
14.         public static void main(String args[])
15.         {
16.             recursion obj = new recursion() ;
17.             System.out.print(obj.fact(6));
18.         }

```

19. }

- a) 1
- b) 30
- c) 120
- d) 720

Answer: d

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
720
```

1. Which of this keyword can be used in a subclass to call the constructor of superclass?

- a) super
- b) this
- c) extent
- d) extends

Answer: a

Explanation: None.

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

- a) Method overloading
- b) Method overriding
- c) Method hiding
- d) None of the mentioned

Answer: b

Explanation: None.

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

- a) static
- b) constant
- c) protected
- d) final

Answer: d

Explanation: 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.

4. Which of these is correct way of calling a constructor having no parameters, of superclass A by subclass B?

- a) super(void);
- b) superclass.();
- c) super.A();
- d) super();

Answer: d

Explanation: None.

5. At line number 2 in the following code, choose 3 valid data-type attributes/qualifiers among “final, static, native, public, private, abstract, protected”

```
1. public interface Status
2.     {
```

```
3.      /* insert qualifier here */ int MY_VALUE = 10;
4.      }
```

- a) final, native, private
- b) final, static, protected
- c) final, private, abstract
- d) final, static, public

Answer: d

Explanation: Every interface variable is implicitly public static and final.

6. Which of these is supported by method overriding in Java?

- a) Abstraction
- b) Encapsulation
- c) Polymorphism
- d) None of the mentioned

Answer: c

Explanation: None.

7. What will be the output of the following Java program?

```
1.  class Alligator
2.  {
3.      public static void main(String[] args)
4.      {
5.          int [][]x[] = {{1,2}, {3,4,5}, {6,7,8,9}};
6.          int [][]y = x;
7.          System.out.println(y[2][1]);
8.      }
9.  }
```

- a) 2
- b) 3
- c) 7
- d) Compilation Error

Answer: c

Explanation: Both x, and y are pointing to the same array.

8. What will be the output of the following Java program?

```
1.      final class A
2.      {
3.          int i;
4.      }
5.      class B extends A
6.      {
```

```

7.         int j;
8.         System.out.println(j + " " + i);
9.     }
10.    class inheritance
11.    {
12.        public static void main(String args[])
13.        {
14.            B obj = new B();
15.            obj.display();
16.        }
17.    }

```

- a) 2 2
- b) 3 3
- c) Runtime Error
- d) Compilation Error

Answer: d

Explanation: class A has been declared final hence it cannot be inherited by any other class. Hence class B does not have member i, giving compilation error.

output:

```

$ javac inheritance.java
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    i cannot be resolved or is not a field

```

9. What will be the output of the following Java program?

```

1.    class Abc
2.    {
3.        public static void main(String[]args)
4.        {
5.            String[] elements = { "for", "tea", "too" };
6.            String first = (elements.length > 0) ? elements[0]: null;
7.        }
8.    }

```

- a) Compilation error
- b) An exception is thrown at run time
- c) The variable first is set to null
- d) The variable first is set to elements[0]

Answer: d

Explanation: The value at the 0th position will be assigned to the variable first.

10. What will be the output of the following Java program?


```

1.      class A
2.      {
3.          int i;
4.          public void display()
5.          {
6.              System.out.println(i);
7.          }
8.      }
9.      class B extends A
10.     {
11.         int j;
12.         public void display()
13.         {
14.             System.out.println(j);
15.         }
16.     }
17.     class Dynamic_dispatch
18.     {
19.         public static void main(String args[])
20.         {
21.             B obj2 = new B();
22.             obj2.i = 1;
23.             obj2.j = 2;
24.             A r;
25.             r = obj2;
26.             r.display();
27.         }
28.     }

```

- a) 1
- b) 2
- c) 3
- d) 4

Answer: b

Explanation: *r* is reference of type A, the program assigns a reference of object *obj2* to *r* and uses that reference to call function *display()* of class B.

output:

```
$ javac Dynamic_dispatch.java  
$ java Dynamic_dispatch  
2
```

1. Which of these class is superclass of every class in Java?

- a) String class**
- b) Object class**
- c) Abstract class**
- d) ArrayList class**

Answer: b

Explanation: Object class is superclass of every class in Java.

2. Which of these method of Object class can clone an object?

- a) Objectcopy()**
- b) copy()**
- c) Object clone()**
- d) clone()**

Answer: c

Explanation: None.

3. Which of these method of Object class is used to obtain class of an object at run time?

- a) get()**
- b) void getclass()**
- c) Class getclass()**
- d) None of the mentioned**

Answer: c

Explanation: None.

4. Which of these keywords can be used to prevent inheritance of a class?

- a) super**
- b) constant**
- c) class**
- d) final**

Answer: d

Explanation: Declaring a class final implicitly declared all of its methods final, and makes the class inheritable.

5. Which of these keywords cannot be used for a class which has been declared final?

- a) abstract**
- b) extends**
- c) abstract and extends**
- d) none of the mentioned**

Answer: a

Explanation: A abstract class is incomplete by itself and relies upon its subclasses to provide a complete implementation. If we declare a class final then no class can inherit that class, an abstract class needs its subclasses hence both final and abstract cannot be used for a same class.

6. Which of these class relies upon its subclasses for complete implementation of its methods?

- a) Object class**
- b) abstract class**
- c) ArrayList class**
- d) None of the mentioned**

Answer: b

Explanation: None.

7. What will be the output of the following Java program?

```
1.      abstract class A
2.      {
3.          int i;
4.          abstract void display();
5.      }
6.      class B extends A
7.      {
8.          int j;
9.          void display()
10.         {
11.             System.out.println(j);
12.         }
13.     }
14.     class Abstract_demo
15.     {
16.         public static void main(String args[])
17.         {
18.             B obj = new B();
19.             obj.j=2;
20.             obj.display();
21.         }
22.     }
```

- a) 0
- b) 2
- c) Runtime Error
- d) Compilation Error

Answer: b

Explanation: class A is an abstract class, it contains a abstract function display(), the full implementation of display() method is given in its subclass B, Both the display functions are the same. Prototype of display() is defined in class A and its implementation is given in class B.

output:

```
$ javac Abstract_demo.java
$ java Abstract_demo
2
```

8. What will be the output of the following Java program?

```

1.      class A
2.      {
3.          int i;
4.          int j;
5.          A()
6.          {
7.              i = 1;
8.              j = 2;
9.          }
10.     }
11.     class Output
12.     {
13.         public static void main(String args[])
14.         {
15.             A obj1 = new A();
16.             A obj2 = new A();
17.             System.out.print(obj1.equals(obj2));
18.         }
19.     }

```

- a) false
- b) true
- c) 1
- d) Compilation Error

Answer: a

Explanation: obj1 and obj2 are two different objects. equals() is a method of Object class, Since Object class is superclass of every class it is available to every object.

output:

```

$ javac Output.java
$ java Output
false

```

9. What will be the output of the following Java code?

```

1.      class Output
2.      {
3.         public static void main(String args[])
4.         {
5.             Object obj = new Object();
6.             System.out.print(obj.getClass());

```

```
7.        }
```

```
8.    }
```

- a) Object
- b) class Object
- c) class java.lang.Object
- d) Compilation Error

Answer: c

Explanation: None.

output:

```
$ javac Output.java
$ java Output
class java.lang.Object
```

10. What will be the output of the following Java code?

```
1.    class A
2.    {
3.        int i;
4.        int j;
5.        A()
6.        {
7.            i = 1;
8.            j = 2;
9.        }
10.   }
11.   class Output
12.   {
13.       public static void main(String args[])
14.       {
15.           A obj1 = new A();
16.           System.out.print(obj1.toString());
17.       }
18.   }
```

- a) true
- b) false
- c) String associated with obj1
- d) Compilation Error

Answer: c

Explanation: toString() is method of class Object, since it is superclass of every class, every object has this method. toString() returns the string associated with the calling object.

output:

```
$ javac Output.java
$ java Output
[email protected]
```

1. Which of these keywords are used to define an abstract class?

- a) abst
- b) abstract
- c) Abstract
- d) abstract class

Answer: b

Explanation: None.

2. Which of these is not abstract?

- a) Thread
- b) AbstractList
- c) List
- d) None of the Mentioned

Answer: a

Explanation: Thread is not an abstract class.

3. If a class inheriting an abstract class does not define all of its function then it will be known as?

- a) Abstract
- b) A simple class
- c) Static class
- d) None of the mentioned

Answer: a

Explanation: Any subclass of an abstract class must either implement all of the abstract method in the superclass or be itself declared abstract.

4. Which of these is not a correct statement?

- a) Every class containing abstract method must be declared abstract
- b) Abstract class defines only the structure of the class not its implementation
- c) Abstract class can be initiated by new operator
- d) Abstract class can be inherited

Answer: c

Explanation: Abstract class cannot be directly initiated with new operator; Since abstract class does not contain any definition of implementation it is not possible to create an abstract object.

5. Which of these packages contains abstract keyword?

- a) java.lang
- b) java.util
- c) java.io
- d) java.system

Answer: a

Explanation: None.

6. What will be the output of the following Java code?

```
1.      class A
2.      {
3.          public int i;
4.          private int j;
```

```

5.      }
6.      class B extends A
7.      {
8.          void display()
9.          {
10.             super.j = super.i + 1;
11.             System.out.println(super.i + " " + super.j);
12.         }
13.     }
14.     class inheritance
15.     {
16.         public static void main(String args[])
17.         {
18.             B obj = new B();
19.             obj.i=1;
20.             obj.j=2;
21.             obj.display();
22.         }
23.     }

```

- a) 2 2
- b) 3 3
- c) Runtime Error
- d) Compilation Error

Answer: d

Explanation: Class contains a private member variable j, this cannot be inherited by subclass B and does not have access to it.

output:

```

$ javac inheritance.java
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    The field A.j is not visible

```

7. What will be the output of the following Java code?

```

1.      class A
2.      {
3.          public int i;
4.          public int j;
5.          A()

```

```

6.      {
7.          i = 1;
8.          j = 2;
9.      }
10.     }
11.     class B extends A
12.     {
13.         int a;
14.         B()
15.         {
16.             super();
17.         }
18.     }
19.     class super_use
20.     {
21.         public static void main(String args[])
22.         {
23.             B obj = new B();
24.             System.out.println(obj.i + " " + obj.j)
25.         }
26.     }

```

- a) 1 2
- b) 2 1
- c) Runtime Error
- d) Compilation Error

Answer: a

Explanation: Keyword super is used to call constructor of class A by constructor of class B. Constructor of a initializes i & j to 1 & 2 respectively.

output:

```

$ javac super_use.java
$ java super_use
1 2

```

8. What will be the output of the following Java code?

```

1.     class A
2.     {
3.         int i;
4.         void display()

```



```

5.      {
6.          System.out.println(i);
7.      }
8.  }
9.  class B extends A
10. {
11.     int j;
12.     void display()
13.     {
14.         System.out.println(j);
15.     }
16. }
17. class method_overriding
18. {
19.     public static void main(String args[])
20.     {
21.         B obj = new B();
22.         obj.i=1;
23.         obj.j=2;
24.         obj.display();
25.     }
26. }

```

- a) 0
- b) 1
- c) 2
- d) Compilation Error

Answer: c

Explanation: class A & class B both contain display() method, class B inherits class A, when display() method is called by object of class B, display() method of class B is executed rather than that of Class A.

output:

```

$ javac method_overriding.java
$ java method_overriding
2

```

9. What will be the output of the following Java code?

```

1.     class A
2.     {
3.         public int i;

```

```

4.         protected int j;
5.     }
6.     class B extends A
7.     {
8.         int j;
9.         void display()
10.        {
11.            super.j = 3;
12.            System.out.println(i + " " + j);
13.        }
14.    }
15.    class Output
16.    {
17.        public static void main(String args[])
18.        {
19.            B obj = new B();
20.            obj.i=1;
21.            obj.j=2;
22.            obj.display();
23.        }
24.    }

```

- a) 1 2
- b) 2 1
- c) 1 3
- d) 3 1

Answer: a

Explanation: Both class A & B have member with same name that is j, member of class B will be called by default if no specifier is used. I contains 1 & j contains 2, printing 1 2.

output:

```

$ javac Output.java
$ java Output
1 2

```

1. Which of this keyword must be used to inherit a class?

- a) super
- b) this
- c) extent
- d) extends

Answer: d

Explanation: None.

2. A class member declared protected becomes a member of subclass of which type?

- a) public member**
- b) private member**
- c) protected member**
- d) static member**

Answer: b

Explanation: A class member declared protected becomes a private member of subclass.

3. Which of these is correct way of inheriting class A by class B?

- a) class B + class A {}**
- b) class B inherits class A {}**
- c) class B extends A {}**
- d) class B extends class A {}**

Answer: c

Explanation: None.

4. Which two classes use the Shape class correctly?

```
A. public class Circle implements Shape
{
    private int radius;
}

B. public abstract class Circle extends Shape
{
    private int radius;
}

C. public class Circle extends Shape
{
    private int radius;
    public void draw();
}

D. public abstract class Circle implements Shape
{
    private int radius;
    public void draw();
}

E. public class Circle extends Shape
{
    private int radius;
    public void draw()
    {
        /* code here */
    }
}

F. public abstract class Circle implements Shape
{
    private int radius;
    public void draw()
    {
        /* code here */
    }
}
```

- a) B,E**
- b) A,C**
- c) C,E**
- d) T,H**

Answer: a

Explanation: If one is extending any class, then they should use extends keyword not implements.

5. What will be the output of the following Java program?

```
1.     class A
2.     {
3.         int i;
4.         void display()
5.         {
6.             System.out.println(i);
7.         }
8.     }
9.     class B extends A
10.    {
11.        int j;
12.        void display()
13.        {
14.            System.out.println(j);
15.        }
16.    }
17.    class inheritance_demo
18.    {
19.        public static void main(String args[])
20.        {
21.            B obj = new B();
22.            obj.i=1;
23.            obj.j=2;
24.            obj.display();
25.        }
26.    }
```

- a) 0
- b) 1
- c) 2
- d) Compilation Error

Answer: c

Explanation: Class A & class B both contain display() method, class B inherits class A, when display() method is called by object of class B, display() method of class B is executed rather than that of Class A.

output:

```
$ javac inheritance_demo.java
$ java inheritance_demo
2
```

6. What will be the output of the following Java program?

```
1.    class A
2.    {
3.        int i;
4.    }
5.    class B extends A
6.    {
7.        int j;
8.        void display()
9.        {
10.            super.i = j + 1;
11.            System.out.println(j + " " + i);
12.        }
13.    }
14.    class inheritance
15.    {
16.        public static void main(String args[])
17.        {
18.            B obj = new B();
19.            obj.i=1;
20.            obj.j=2;
21.            obj.display();
22.        }
23.    }
```

- a) 2 2
- b) 3 3
- c) 2 3
- d) 3 2

Answer: c

Explanation: None

output:

```
$ javac inheritance.java
$ java inheritance
2 3
```

7. What will be the output of the following Java program?

```
1.    class A
```

```

2.      {
3.          public int i;
4.          public int j;
5.          A()
6.          {
7.              i = 1;
8.              j = 2;
9.          }
10.     }
11.     class B extends A
12.     {
13.         int a;
14.         B()
15.         {
16.             super();
17.         }
18.     }
19.     class super_use
20.     {
21.         public static void main(String args[])
22.         {
23.             B obj = new B();
24.             System.out.println(obj.i + " " + obj.j)
25.         }
26.     }

```

- a) 1 2
- b) 2 1
- c) Runtime Error
- d) Compilation Error

Answer: a

Explanation: Keyword super is used to call constructor of class A by constructor of class B. Constructor of a initializes i & j to 1 & 2 respectively.

output:

```

$ javac super_use.java
$ java super_use
1 2

```

1. What is not type of inheritance?

- a) Single inheritance
- b) Double inheritance
- c) Hierarchical inheritance
- d) Multiple inheritance

Answer: b

Explanation: Inheritance is way of acquiring attributes and methods of parent class. Java supports hierarchical inheritance directly.

2. Using which of the following, multiple inheritance in Java can be implemented?

- a) Interfaces
- b) Multithreading
- c) Protected methods
- d) Private methods

Answer: a

Explanation: Multiple inheritance in java is implemented using interfaces. Multiple interfaces can be implemented by a class.

3. All classes in Java are inherited from which class?

- a) java.lang.class
- b) java.class.inherited
- c) java.class.object
- d) java.lang.Object

Answer: d

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

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

- a) Protected
- b) Private
- c) Public
- d) Static

Answer: b

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

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

- a) super
- b) this
- c) upper
- d) classname

Answer: a

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

Answer: b
Explanation: Static members are also inherited to subclasses.

7. Which of the following is used for implementing inheritance through an interface?

- a) inherited
- b) using
- c) extends
- d) implements

Answer: d

Explanation: Interface is implemented using implements keyword. A concrete class must implement all the methods of an interface, else it must be declared abstract.

8. Which of the following is used for implementing inheritance through class?

- a) inherited
- b) using
- c) extends
- d) implements

Answer: c

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

**9. What would be the result if a class extends two interfaces and both have a method with same name and signature?
Let's assume that the class is not implementing that method.**

- a) Runtime error
- b) Compile time error
- c) Code runs successfully
- d) First called method is executed successfully

Answer: b

Explanation: In case of such conflict, compiler will not be able to link a method call due to ambiguity. It will throw compile time error.

Answer: a

Explanation: Java supports multiple level inheritance through implementing multiple interfaces.

1. Which of these class is superclass of String and StringBuffer class?

- a) java.util
- b) java.lang
- c) ArrayList
- d) None of the mentioned

Answer: b

Explanation: None.

2. Which of these operators can be used to concatenate two or more String objects?

- a) +
- b) +=
- c) &
- d) ||

Answer: a

Explanation: Operator + is used to concatenate strings, Example String s = "i " + "like " + "java"; String s contains "I like java".

3. Which of this method of class String is used to obtain a length of String object?

- a) get()
- b) Sizeof()
- c) lengthof()
- d) length()

Answer: d

Explanation: Method length() of string class is used to get the length of the object which invoked method length().

4. Which of these method of class String is used to extract a single character from a String object?

- a) CHARAT()
- b) chatat()
- c) charAt()
- d) ChatAt()

Answer: c

Explanation: None.

5. Which of these constructors is used to create an empty String object?

- a) String()
- b) String(void)
- c) String(0)
- d) None of the mentioned

Answer: a

Explanation: None.

6. Which of these is an incorrect statement?

- a) String objects are immutable, they cannot be changed
- b) String object can point to some other reference of String variable
- c) StringBuffer class is used to store string in a buffer for later use
- d) None of the mentioned

Answer: c

Explanation: StringBuffer class is used to create strings that can be modified after they are created.

7. What will be the output of the following Java program?

```
1.      class String_demo
2.      {
3.          public static void main(String args[])
4.          {
5.              char chars[] = {'a', 'b', 'c'};
6.              String s = new String(chars);
7.              System.out.println(s);
8.          }
9.      }
```

- a) a
- b) b
- c) c
- d) abc

Answer: d

Explanation: String(chars) is a constructor of class string, it initializes string s with the values stored in character array chars, therefore s contains “abc”.

output:

```
$ javac String_demo.java
$ java String_demo
abc
```

8. What will be the output of the following Java program?

```
1.      class String_demo
2.      {
3.          public static void main(String args[])
4.          {
```

```

5.         int ascii[] = { 65, 66, 67, 68};
6.
7.         String s = new String(ascii, 1, 3);
8.
9.         System.out.println(s);
10.
11.    }
12.
13. }

```

- a) ABC
- b) BCD
- c) CDA
- d) ABCD

Answer: b

Explanation: *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*.

output:

```

$ javac String_demo.java
$ java String_demo
BCD

```

9. What will be the output of the following Java program?

```

1.    class String_demo
2.    {
3.
4.        public static void main(String args[])
5.        {
6.
7.            char chars[] = {'a', 'b', 'c'};
8.
9.            String s = new String(chars);
10.
11.           String s1 = "abcd";
12.
13.           int len1 = s1.length();
14.
15.           int len2 = s.length();
16.
17.           System.out.println(len1 + " " + len2);
18.
19.        }
20.    }

```

- a) 3 0
- b) 0 3
- c) 3 4
- d) 4 3

Answer: d

Explanation: None.

output:

```

$ javac String_demo.java
$ java String_demo
4 3

```

1. Which of these method of class String is used to extract more than one character at a time a String object?

- a) getchars()
- b) GetChars()
- c) Getchars()
- d) getChars()

Answer: d

Explanation: None.

2. Which of these methods is an alternative to getChars() that stores the characters in an array of bytes?

- a) getBytes()
- b) GetByte()
- c) giveByte()
- d) Give Bytes()

Answer: a

Explanation: getBytes() stores the character in an array of bytes. It uses default character to byte conversions provided by the platform.

3. In the following Java code, what can directly access and change the value of the variable name?

```
1. package test;
2. class Target
3. {
4.     public String name = "hello";
5. }
```

- a) any class
- b) only the Target class
- c) any class in the test package
- d) any class that extends Target

Answer: c

Explanation: Any class in the test package can access and change name.

4. What will be the output of the following Java code?

```
1. public class Boxer1
2. {
3.     Integer i;
4.     int x;
5.     public Boxer1(int y)
6.     {
7.         x = i+y;
8.         System.out.println(x);
9.     }
10.     public static void main(String[] args)
11.     {
```

```
12.         new Boxer1 (new Integer(4));
13.     }
14. }
```

- a) The value “4” is printed at the command line
- b) Compilation fails because of an error in line
- c) A NullPointerException occurs at runtime
- d) An IllegalStateException occurs at runtime

Answer: d

Explanation: Because we are performing operation on reference variable which is null.

5. Which of these methods can be used to convert all characters in a String into a character array?

- a) charAt()
- b) both getChars() & charAt()
- c) both toCharArray() & getChars()
- d) all of the mentioned

Answer: c

Explanation: charAt() return one character only not array of character.

6. What will be the output of the following Java code?

```
1.     class output
2.     {
3.         public static void main(String args[])
4.         {
5.             String c = "Hello i love java";
6.             int start = 2;
7.             int end = 9;
8.             char s[]=new char[end-start];
9.             c.getChars(start,end,s,0);
10.            System.out.println(s);
11.        }
12.    }
```

- a) Hello, i love java
- b) i love ja
- c) lo i lo
- d) llo i l

Answer: d

Explanation: getChars(start,end,s,0) returns an array from the string c, starting index of array is pointed by start and ending index is pointed by end. s is the target character array where the new string of letters is going to be stored and the new string will be stored from 0th position in s.

Output:

```
$ javac output.java
$ java output
```

7. What will be the output of the following Java code?

```

1.    class output
2.    {
3.        public static void main(String args[])
4.        {
5.            String a = "hello i love java";
6.            System.out.println(a.indexOf('i')+" "+a.indexOf('o') +" "+a.lastIndexOf('i')+" "+a.lastIndexOf('o'));
7.        }
8.    }

```

a) 6 4 6 9

b) 5 4 5 9

c) 7 8 8 9

d) 4 3 6 9

Answer: a

Explanation: indexOf('c') and lastIndexOf('c') are pre defined function which are used to get the index of first and last occurrence of

the character pointed by c in the given array.

Output:

```

$ javac output.java
$ java output
6 4 6 9

```

8. What will be the output of the following Java code?

```

1.    class output
2.    {
3.        public static void main(String args[])
4.        {
5.            char c[]={'a', 'l', 'b' , ' ' , 'A' , '0'};
6.            for (int i = 0; i < 5; ++i)
7.            {
8.                if(Character.isDigit(c[i]))
9.                    System.out.println(c[i]+" is a digit");
10.               if(Character.isWhitespace(c[i]))
11.                   System.out.println(c[i]+" is a Whitespace character");
12.               if(Character.isUpperCase(c[i]))
13.                   System.out.println(c[i]+" is an Upper case Letter");
14.               if(Character.isLowerCase(c[i]))

```

```
15.                System.out.println(c[i]+" is a lower case Letter");
16.                i=i+3;
17.            }
18.        }
19.    }
```

9. What will be the output of the following Java code?

```
1.    class output
2.    {
3.        public static void main(String args[])
4.        {
5.            char ch;
6.            ch = "hello".charAt(1);
7.            System.out.println(ch);
8.        }
9.    }
```

- a) h
- b) e
- c) l
- d) o

Answer: b

Explanation: "hello" is a String literal, method charAt() returns the character specified at the index position. Character at index position 1 is e of hello, hence ch contains e.

output:

```
$ javac output.java
$ java output
e
```

1. Which of these method of class String is used to compare two String objects for their equality?

- a) equals()
- b) Equals()
- c) isequal()
- d) Isequal()

Answer: a

Explanation: None.

2. Which of these methods is used to compare a specific region inside a string with another specific region in another string?

- a) regionMatch()
- b) match()
- c) RegionMatches()
- d) regionMatches()

Answer: d

Explanation: None.

3. Which of these methods of class String is used to check whether a given object starts with a particular string literal?
- a) startsWith()
 - b) endsWith()
 - c) Starts()
 - d) ends()

Answer: a

Explanation: Method startsWith() of string class is used to check whether the String in question starts with a specified string. It is a specialized form of method regionMatches().

4. What is the value returned by function compareTo() if the invoking string is less than the string compared?
- a) zero
 - b) value less than zero
 - c) value greater than zero
 - d) none of the mentioned

Answer: b

Explanation: compareTo() function returns zero when both the strings are equal, it returns a value less than zero if the invoking string is less than the other string being compared and value greater than zero when invoking string is greater than the string compared to.

5. Which of these data type value is returned by equals() method of String class?
- a) char
 - b) int
 - c) boolean
 - d) all of the mentioned

Answer: c

Explanation: equals() method of string class returns boolean value true if both the string are equal and false if they are unequal.

6. What will be the output of the following Java code?

```
1.      class output
2.      {
3.          public static void main(String args[])
4.          {
5.              String c = "Hello i love java";
6.              boolean var;
7.              var = c.startsWith("hello");
8.              System.out.println(var);
9.          }
10.     }
```

- a) true
- b) false
- c) 0
- d) 1

Answer: b

Explanation: startsWith() method is case sensitive “hello” and “Hello” are treated differently, hence false is stored in

var:

Output:

```
$ javac output.java
$ java output
false
```

7. What will be the output of the following Java code?

```
1.      class output
2.      {
3.          public static void main(String args[])
4.          {
5.              String s1 = "Hello i love java";
6.              String s2 = new String(s1);
7.              System.out.println((s1 == s2) + " " + s1.equals(s2));
8.          }
9.      }
```

- a) true true
- b) false false
- c) true false
- d) false true

Answer: d
Explanation: The == operator compares two object references to see whether they refer to the same instance, where as equals() compares the content of the two objects.

Output:

```
$ javac output.java
$ java output
false true
```

8. What will be the output of the following Java code?

```
1.      class output
2.      {
3.          public static void main(String args[])
4.          {
5.              String s1 = "Hello";
6.              String s2 = new String(s1);
7.              String s3 = "HELLO";
8.              System.out.println(s1.equals(s2) + " " + s2.equals(s3));
9.          }
10.     }
```

- a) true true

- b) false false
- c) true false
- d) false true

Answer: c

Explanation: None.

Output:

```
$ javac output.java
$ java output
true false
```

9. In the following Java code, which code fragment should be inserted at line 3 so that the output will be: “123abc 123abc”?

```
1. StringBuilder sb1 = new StringBuilder("123");
2. String s1 = "123";
3. // insert code here
4. System.out.println(sb1 + " " + s1);
```

- a) sb1.append(“abc”); s1.append(“abc”);
- b) sb1.append(“abc”); s1.concat(“abc”);
- c) sb1.concat(“abc”); s1.append(“abc”);
- d) sb1.append(“abc”); s1 = s1.concat(“abc”);

Answer: d

Explanation: *append()* is stringbuffer method and *concat* is String class method.

append() is stringbuffer method and *concat* is String class method.

10. What will be the output of the following Java code?

```
1. class output
2. {
3.     public static void main(String args[])
4.     {
5.         String chars[] = {"a", "b", "c", "a", "c"};
6.         for (int i = 0; i < chars.length; ++i)
7.             for (int j = i + 1; j < chars.length; ++j)
8.                 if(chars[i].compareTo(chars[j]) == 0)
9.                     System.out.print(chars[j]);
10.    }
11. }
```

- a) ab
- b) bc
- c) ca
- d) ac

Answer: d

Explanation: *compareTo()* function returns zero when both the strings are equal, it returns a value less than zero if the invoking string is less than the other string being compared and value greater than zero when invoking string is greater than the string compared to.

output:

```
$ javac output.java
$ java output
ac
```

1. Which of this method of class String is used to extract a substring from a String object?

- a) substring()**
- b) Substring()**
- c) SubString()**
- d) None of the mentioned**

Answer: a

Explanation: None.

2. What will s2 contain after following lines of Java code?

```
String s1 = "one";
String s2 = s1.concat("two")
```

- a) one**
- b) two**
- c) onetwo**
- d) twoone**

Answer: c

Explanation: Two strings can be concatenated by using concat() method.

3. Which of these method of class String is used to remove leading and trailing whitespaces?

- a) startsWith()**
- b) trim()**
- c) Trim()**
- d) doTrim()**

Answer: b

Explanation: None.

4. What is the value returned by function compareTo() if the invoking string is greater than the string compared?

- a) zero**
- b) value less than zero**
- c) value greater than zero**
- d) none of the mentioned**

Answer: c

Explanation:

```
if (s1 == s2) then 0, if(s1 > s2) > 0, if (s1 < s2) then < 0.
```

5. Which of the following statement is correct?

- a) replace() method replaces all occurrences of one character in invoking string with another character**
- b) replace() method replaces only first occurrences of a character in invoking string with another character**
- c) replace() method replaces all the characters in invoking string with another character**
- d) replace() method replaces last occurrence of a character in invoking string with another character**

Answer: a

Explanation: replace() method replaces all occurrences of one character in invoking string with another character.

6. What will be the output of the following Java program?

```
1.      class output
```

```

2.      {
3.          public static void main(String args[])
4.      {
5.          String c = "  Hello World  ";
6.          String s = c.trim();
7.          System.out.println("\""+s+"\"");
8.      }
9.  }

```

- a) ""Hello World""
- b) ""Hello World"
- c) "Hello World"
- d) Hello world

Answer: c

Explanation: trim() method is used to remove leading and trailing whitespaces in a string.

Output:

```

$ javac output.java
$ java output
"Hello World"

```

7. What will be the output of the following Java program?

```

1.      class output
2.      {
3.          public static void main(String args[])
4.      {
5.          String s1 = "one";
6.          String s2 = s1 + " two";
7.          System.out.println(s2);
8.      }
9.  }

```

- a) one
- b) two
- c) one two
- d) compilation error

Answer: c

Explanation: None.

Output:

```

$ javac output.java
$ java output
one two

```

8. What will be the output of the following Java program?

```

1.     class output
2.     {
3.         public static void main(String args[])
4.         {
5.             String s1 = "Hello";
6.             String s2 = s1.replace('l','w');
7.             System.out.println(s2);
8.         }
9.     }

```

- a) hello
- b) helwo
- c) hewlo
- d) hewwo

Answer: d

Explanation: replace() method replaces all occurrences of one character in invoking string with another character. s1.replace('l','w') replaces every occurrence of 'l' in hello by 'w', giving hewwo.

Output:

```

$ javac output.java
$ java output
hewwo

```

9. What will be the output of the following Java program?

```

1.     class output
2.     {
3.         public static void main(String args[])
4.         {
5.             String s1 = "Hello World";
6.             String s2 = s1.substring(0 , 4);
7.             System.out.println(s2);
8.         }
9.     }

```

- a) Hell
- b) Hello
- c) Worl
- d) World

Answer: a

Explanation: substring(0,4) returns the character from 0 th position to 3 rd position.

output:

```

$ javac output.java
$ java output
Hell

```

10. What will be the output of the following Java program?

```
1.      class output
2.      {
3.          public static void main(String args[])
4.          {
5.              String s = "Hello World";
6.              int i = s.indexOf('o');
7.              int j = s.lastIndexOf('l');
8.              System.out.print(i + " " + j);
9.          }
10.     }
```

- a) 4 8**
- b) 5 9**
- c) 4 9**
- d) 5 8**

Answer: c
Explanation: indexOf() method returns the index of first occurrence of the character where as lastIndexOf() returns the index of last occurrence of the character.
output:

```
$ javac output.java
$ java output
4 9
```

1. Which of these class is used to create an object whose character sequence is mutable?

- a) String()**
- b) StringBuffer()**
- c) String() & StringBuffer()**
- d) None of the mentioned**

Answer: b
Explanation: StringBuffer represents growable and writable character sequence.

2. Which of this method of class StringBuffer is used to concatenate the string representation to the end of invoking string?

- a) concat()**
- b) append()**
- c) join()**
- d) concatenate()**

Answer: b
Explanation: None.

3. Which of these method of class StringBuffer is used to find the length of current character sequence?

- a) length()**
- b) Length()**
- c) capacity()**
- d) Capacity()**

Answer: a
Explanation: None.

4. What is the string contained in s after following lines of Java code?

```
StringBuffer s = new StringBuffer("Hello");  
s.deleteCharAt(0);
```

- a) Hell
- b) ello
- c) Hel
- d) llo

Answer: b
Explanation: deleteCharAt() method deletes the character at the specified index location and returns the resulting StringBuffer object.

5. Which of the following statement is correct?

- a) reverse() method reverses all characters
- b) reverseall() method reverses all characters
- c) replace() method replaces first occurrence of a character in invoking string with another character
- d) replace() method replaces last occurrence of a character in invoking string with another character

Answer: a
Explanation: reverse() method reverses all characters. It returns the reversed object on which it was called.

6. What will be the output of the following Java program?

```
1.      class output  
2.      {  
3.          public static void main(String args[])  
4.          {  
5.              String a = "hello i love java";  
6.              System.out.println(a.indexOf('e')+" "+a.indexOf('a')+" "+a.lastIndexOf('l')+" "+a  
7.          }  
8.      }
```

- a) 6 4 6 9
- b) 5 4 5 9
- c) 7 8 8 9
- d) 1 14 8 15

Answer: d
Explanation: indexOf('c') and lastIndexof('c') are pre defined function which are used to get the index of first and last occurrence of the character pointed by c in the given array.
Output:

```
$ javac output.java  
$ java output  
1 14 8 15
```

7. What will be the output of the following Java program?

```
1.      class output
```

```

2.      {
3.          public static void main(String args[])
4.      {
5.          StringBuffer c = new StringBuffer("Hello");
6.          c.delete(0,2);
7.          System.out.println(c);
8.      }
9.  }

```

- a) He
- b) Hel
- c) lo
- d) llo

Answer: d

Explanation: delete(0,2) is used to delete the characters from 0 th position to 1 st position.

Output:

```

$ javac output.java
$ java output
llo

```

8. What will be the output of the following Java program?

```

1.      class output
2.      {
3.          public static void main(String args[])
4.      {
5.          StringBuffer c = new StringBuffer("Hello");
6.          StringBuffer c1 = new StringBuffer(" World");
7.          c.append(c1);
8.          System.out.println(c);
9.      }
10.     }

```

- a) Hello
- b) World
- c) Helloworld
- d) Hello World

Answer: d

Explanation: append() method of class StringBuffer is used to concatenate the string representation to the end of invoking string.

Output:

```

$ javac output.java
$ java output

```

9. What will be the output of the following Java program?

```

1.    class output
2.    {
3.        public static void main(String args[])
4.        {
5.            StringBuffer s1 = new StringBuffer("Hello");
6.            StringBuffer s2 = s1.reverse();
7.            System.out.println(s2);
8.        }
9.    }

```

- a) Hello
- b) olleH
- c) HelloolleH
- d) olleHHello

Answer: b

Explanation: reverse() method reverses all characters. It returns the reversed object on which it was called.

Output:

```

$ javac output.java
$ java output
olleH

```

10. What will be the output of the following Java program?

```

1.    class output
2.    {
3.        class output
4.        {
5.            public static void main(String args[])
6.            {
7.                char c[]={ 'A', '1', 'b' , ' ' , 'a' , '0' };
8.                for (int i = 0; i < 5; ++i)
9.                {
10.                    i++;
11.                    if(Character.isDigit(c[i]))
12.                        System.out.println(c[i]+" is a digit");
13.                    if(Character.isWhitespace(c[i]))
14.                        System.out.println(c[i]+" is a Whitespace character");

```



```

15.                if(Character.isUpperCase(c[i]))
16.                    System.out.println(c[i]+" is an Upper case Letter");
17.                if(Character.isLowerCase(c[i]))
18.                    System.out.println(c[i]+" is a lower case Letter");
19.                i++;
20.            }
21.        }
22.    }

```

1. Which of these methods of class StringBuffer is used to extract a substring from a String object?

- a) substring()
- b) Substring()
- c) SubString()
- d) None of the mentioned

Answer: a

Explanation: None.

2. What will s2 contain after following lines of Java code?

```

StringBuffer s1 = "one";
StringBuffer s2 = s1.append("two")

```

- a) one
- b) two
- c) onetwo
- d) twoone

Answer: c

Explanation: Two strings can be concatenated by using append() method.

3. Which of this method of class StringBuffer is used to reverse sequence of characters?

- a) reverse()
- b) reverseall()
- c) Reverse()
- d) reverseAll()

Answer: a

Explanation: reverse() method reverses all characters. It returns the reversed object on which it was called.

4. Which of this method of class StringBuffer is used to get the length of the sequence of characters?

- a) length()
- b) capacity()
- c) Length()
- d) Capacity()

Answer: a

Explanation: length()- returns the length of String the StringBuffer would create whereas capacity() returns a total number of characters that can be supported before it is grown.

5. Which of the following are incorrect form of StringBuffer class constructor?

- a) StringBuffer()
- b) StringBuffer(int size)
- c) StringBuffer(String str)

d) StringBuffer(int size , String str)

Answer: d

Explanation: None.

6. What will be the output of the following Java code?

```
1.  class output
2.  {
3.      public static void main(String args[])
4.      {
5.          StringBuffer c = new StringBuffer("Hello");
6.          System.out.println(c.length());
7.      }
8.  }
```

a) 4

b) 5

c) 6

d) 7

Answer: b

Explanation: length() method is used to obtain length of StringBuffer object, length of “Hello” is 5.

Output:

```
$ javac output.java
$ java output
5
```

7. What will be the output of the following Java code?

```
1.  class output
2.  {
3.      public static void main(String args[])
4.      {
5.          StringBuffer sb=new StringBuffer("Hello");
6.          sb.replace(1,3,"Java");
7.          System.out.println(sb);
8.      }
9.  }
```

a) Hello java

b) Hellojava

c) HJavallo

d) Hjava

Answer: c

Explanation: The replace() method replaces the given string from the specified beginIndex and endIndex.

```
$ javac output.java
$ java output
HJavallo
```

8. What will be the output of the following Java code?

```
1.    class output
2.    {
3.        public static void main(String args[])
4.        {
5.            StringBuffer s1 = new StringBuffer("Hello");
6.            s1.setCharAt(1, 'x');
7.            System.out.println(s1);
8.        }
9.    }
```

- a) xello
- b) xxxxx
- c) Hxlllo
- d) Hexlo

Answer: c

Explanation: None.

Output:

```
$ javac output.java
$ java output
Hxlllo
```

9. What will be the output of the following Java code?

```
1.    class output
2.    {
3.        public static void main(String args[])
4.        {
5.            StringBuffer s1 = new StringBuffer("Hello World");
6.            s1.insert(6 , "Good ");
7.            System.out.println(s1);
8.        }
9.    }
```

- a) HelloGoodWorld
- b) HellGoodoWorld
- c) HellGood oWorld
- d) Hello Good World

Answer: d

Explanation: The insert() method inserts one string into another. It is overloaded to accept values of all simple types,

plus String and Objects. Sting is inserted into invoking object at specified position. “Good ” is inserted in “Hello World” T index 6 giving “Hello Good World”.

output:

```
$ javac output.java
$ java output
Hello Good World
```

10. What will be the output of the following Java code?

```
1.      class output
2.      {
3.          public static void main(String args[])
4.          {
5.              StringBuffer s1 = new StringBuffer("Hello");
6.              s1.insert(1,"Java");
7.              System.out.println(s1);
8.          }
9.      }
```

- a) hello
- b) java
- c) Hello Java
- d) HJavaello

Answer: d

Explanation: Insert method will insert string at a specified position

Output:

```
$ javac output.java
$ java output
HJavaello
```

1. Which of these classes is not included in java.lang?

- a) Byte
- b) Integer
- c) Array
- d) Class

Answer: c

Explanation: Array class is a member of java.util.

2. Which of these is a process of converting a simple data type into a class?

- a) type wrapping
- b) type conversion
- c) type casting
- d) none of the Mentioned

Answer: a

Explanation: None.

3. Which of these is a super class of wrappers Double & Integer?

- a) Long
- b) Digits

- c) Float
- d) Number

Answer: d

Explanation: Number is an abstract class containing subclasses Double, Float, Byte, Short, Integer and Long.

4. Which of these is a wrapper for simple data type float?

- a) float
- b) double
- c) Float
- d) Double

Answer: c

Explanation: None.

5. Which of the following is a method of wrapper Float for converting the value of an object into byte?

- a) bytevalue()
- b) byte byteValue()
- c) Bytevalue()
- d) Byte Bytevalue()

Answer: b

Explanation: None.

6. Which of these methods is used to check for infinitely large and small values?

- a) isInfinite()
- b) isNaN()
- c) Isinfinite()
- d) IsNaN()

Answer: a

Explanation: isinfinite() method returns true is the value being tested is infinitely large or small in magnitude.

7. Which of the following package stores all the simple data types in java?

- a) lang
- b) java
- c) util
- d) java.packages

Answer: a

Explanation: None.

8. What will be the output of the following Java code?

```
1.    class isinfinite_output
2.    {
3.        public static void main(String args[])
4.        {
5.            Double d = new Double(1 / 0.);
6.            boolean x = d.isInfinite();
7.            System.out.print(x);
8.        }
9.    }
```

- a) 0
- b) 1
- c) true
- d) false

Answer: c

Explanation: `isInfinite()` method returns true if the value being tested is infinitely large or small in magnitude. `1/0.` is infinitely large in magnitude hence true is stored in `x`.

Output:

```
$ javac isinfinite_output.java
$ java isinfinite_output
true
```

9. What will be the output of the following Java code?

```
1. class isNaN_output
2. {
3.     public static void main(String args[])
4.     {
5.         Double d = new Double(1 / 0.);
6.         boolean x = d.isNaN();
7.         System.out.print(x);
8.     }
9. }
```

- a) 0
- b) 1
- c) true
- d) false

Answer: d

Explanation: `isNaN()` method returns true if the value being tested is a number. `1/0.` is infinitely large in magnitude, which cannot be defined as a number hence false is stored in `x`.

Output:

```
$ javac isNaN_output.java
$ java isNaN_output
false
```

10. What will be the output of the following Java code?

```
1. class binary
2. {
3.     public static void main(String args[])
4.     {
5.         int num = 17;
6.         System.out.print(Integer.toBinaryString(num));
7.     }
```

8. }

- a) 1001
- b) 10011
- c) 11011
- d) 10001

Answer: d

Explanation: None.

output:

```
$ javac binary.java
$ java binary
10001
```

1. Which of these is a wrapper for data type int?

- a) Integer
- b) Long
- c) Byte
- d) Double

Answer: a

Explanation: None.

2. Which of the following methods is a method of wrapper Integer for obtaining hash code for the invoking object?

- a) int hash()
- b) int hashCode()
- c) int hashCode()
- d) Integer hashCode()

Answer: c

Explanation: None.

3. Which of these is a super class of wrappers Long, Character & Integer?

- a) Long
- b) Digits
- c) Float
- d) Number

Answer: d

Explanation: Number is an abstract class containing subclasses Double, Float, Byte, Short, Integer and Long.

4. Which of these is a wrapper for simple data type char?

- a) Float
- b) Character
- c) String
- d) Integer

Answer: b

Explanation: None.

5. Which of the following is method of wrapper Integer for converting the value of an object into int?

- a) bytevalue()
- b) int intValue();
- c) Bytevalue()
- d) Byte Bytevalue()

Answer: b

Explanation: None.

6. Which of these methods is used to obtain value of invoking object as a long?

- a) long value()
- b) long longValue()
- c) Long longvalue()
- d) Long Longvalue()

Answer: b

Explanation: long longValue() is used to obtain value of invoking object as a long.

7. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              char a[] = {'a', '5', 'A', ' '};
6.              System.out.print(Character.isDigit(a[0]) + " ");
7.              System.out.print(Character.isWhitespace(a[3]) + " ");
8.              System.out.print(Character.isUpperCase(a[2]));
9.          }
10.     }
```

- a) true false true
- b) false true true
- c) true true false
- d) false false false

Answer: b

Explanation: Character.isDigit(a[0]) checks for a[0], whether it is a digit or not, since a[0] i.e 'a' is a character false is returned. a[3] is a whitespace hence Character.isWhitespace(a[3]) returns a true. a[2] is an uppercase letter i.e 'A' hence Character.isUpperCase(a[2]) returns true.

Output:

```
$ javac Output.java
$ java Output
false true true
```

8. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              Integer i = new Integer(257);
6.              byte x = i.byteValue();
7.              System.out.print(x);
8.          }
```



```
9.      }
```

- a) 0
- b) 1
- c) 256
- d) 257

Answer: b

Explanation: `i.byteValue()` method returns the value of wrapper `i` as a byte value. `i` is 257, range of byte is 256 therefore `i` value exceeds byte range by 1 hence 1 is returned and stored in `x`.

Output:

```
$ javac Output.java
$ java Output
1
```

9. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              Integer i = new Integer(257);
6.              float x = i.floatValue();
7.              System.out.print(x);
8.          }
9.      }
```

- a) 0
- b) 1
- c) 257
- d) 257.0

Answer: d

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
257.0
```

10. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              Long i = new Long(256);
6.              System.out.print(i.hashCode());
```

7. }

8. }

- a) 256
- b) 256.0
- c) 256.00
- d) 257.00

Answer: a
Explanation: None.
Output:

```
$ javac Output.java
$ java Output
256
```

1. Which of these class have only one field ‘TYPE’?
- a) Void
 - b) Process
 - c) System
 - d) Runtime

Answer: a
Explanation: The Void class has one field, TYPE, which holds a reference to the Class object for the type void.

2. Which of the following method of Process class can terminate a process?
- a) void kill()
 - b) void destroy()
 - c) void terminate()
 - d) void exit()

Answer: b
Explanation: Kills the subprocess. The subprocess represented by this Process object is forcibly terminated.

3. Standard output variable ‘out’ is defined in which class?
- a) Void
 - b) Process
 - c) Runtime
 - d) System

Answer: d
Explanation: Standard output variable ‘out’ is defined in System class. out is usually used in print statement i.e System.out.print().

4. Which of these class can encapsulate an entire executing program?
- a) Void
 - b) Process
 - c) Runtime
 - d) System

Answer: b
Explanation: None.

5. Which of the following is method of System class is used to find how long a program takes to execute?
- a) currenttime()
 - b) currentTime()
 - c) currentTimeMillis()
 - d) currenttimeMillis()

Answer: c

Explanation: None.

6. Which of these class holds a collection of static methods and variables?

- a) Void**
- b) Process**
- c) Runtime**
- d) System**

Answer: d

Explanation: System class holds a collection of static methods and variables. The standard input, output and error output of java runtime is stored in the in, out and err variables of System class.

7. What will be the output of the following Java code?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              long start, end;
6.              start = System.currentTimeMillis();
7.              for (int i = 0; i < 100000000; i++);
8.              end = System.currentTimeMillis();
9.              System.out.print(end - start);
10.         }
11.     }
```

- a) 0**
- b) 1**
- c) 1000**
- d) System Dependent**

Answer: d

Explanation: end time is the time taken by loop to execute it can be any non zero value depending on the System.

Output:

```
$ javac Output.java
$ java Output
78
```

8. What will be the output of the following Java code?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              byte a[] = { 65, 66, 67, 68, 69, 70 };
6.              byte b[] = { 71, 72, 73, 74, 75, 76 };
```

```

7.         System.arraycopy(a , 0, b, 0, a.length);
8.         System.out.print(new String(a) + " " + new String(b));
9.     }
10. }

```

- a) ABCDEF ABCDEF
- b) ABCDEF GHIJKL
- c) GHIJKL ABCDEF
- d) GHIJKL GHIJKL

Answer: a

Explanation: System.arraycopy() is a method of class System which is used to copy a string into another string.

Output:

```

$ javac Output.java
$ java Output
ABCDEF ABCDEF

```

9. What will be the output of the following Java code?

```

1.     class Output
2.     {
3.         public static void main(String args[])
4.         {
5.             byte a[] = { 65, 66, 67, 68, 69, 70 };
6.             byte b[] = { 71, 72, 73, 74, 75, 76 };
7.             System.arraycopy(a, 2, b, 1, a.length-2);
8.             System.out.print(new String(a) + " " + new String(b));
9.         }
10.    }

```

- a) ABCDEF GHIJKL
- b) ABCDEF GCDEFL
- c) GHIJKL ABCDEF
- d) GCDEFL GHIJKL

Answer: b

Explanation: None.

Output:

```

$ javac Output.java
$ java Output
ABCDEF GCDEFL

```

10. What will be the output of the following Java code?

```

1.     class Output
2.     {
3.         public static void main(String args[])

```

```

4.      {
5.          byte a[] = { 65, 66, 67, 68, 69, 70 };
6.          byte b[] = { 71, 72, 73, 74, 75, 76 };
7.          System.arraycopy(a, 1, b, 3, 0);
8.          System.out.print(new String(a) + " " + new String(b));
9.      }
10.     }

```

- a) ABCDEF GHIJKL
- b) ABCDEF GCDEFL
- c) GHIJKL ABCDEF
- d) GCDEFL GHIJKL

Answer: a

Explanation: Since last parameter of System.arraycopy(a,1,b,3,0) is 0 nothing is copied from array a to array b, hence b remains as it is.

Output:

```

$ javac Output.java
$ java Output
ABCDEF GHIJKL

```

1. Which of these class is a superclass of all other classes?

- a) Math
- b) Process
- c) System
- d) Object

Answer: d

Explanation: The object class class is a superclass of all other classes.

2. Which of these method of Object class can generate duplicate copy of the object on which it is called?

- a) clone()
- b) copy()
- c) duplicate()
- d) dito()

Answer: a

Explanation: None.

3. What is the value of double constant 'E' defined in Math class?

- a) approximately 3
- b) approximately 3.14
- c) approximately 2.72
- d) approximately 0

Answer: c

Explanation: None.

4. Which of these method is a rounding function of Math class?

- a) max()
- b) min()
- c) abs()
- d) all of the mentioned

Answer: d

Explanation: max(), min() and abs() are all rounding functions.

5. Which of these class contains only floating point functions?

- a) Math**
- b) Process**
- c) System**
- d) Object**

Answer: a

Explanation: Math class contains all the floating point functions that are used for geometry, trigonometry, as well as several general purpose methods. Example : sin(), cos(), exp(), sqrt() etc.

6. Which of these class encapsulate the runtime state of an object or an interface?

- a) Class**
- b) Object**
- c) Runtime**
- d) System**

Answer: a

Explanation: None.

7. What is the value of “d” in the following Java code snippet?

```
double d = Math.round ( 2.5 + Math.random() );
```

- a) 2**
- b) 3**
- c) 4**
- d) 2.5**

Answer: b

Explanation: The Math.random() method returns a number greater than or equal to 0 and less than 1. so 2.5 will be greater than or equal to 2.5 and less than 3.5, we can be sure that Math.round() will round that number to 3.

8. What will be the output of the following Java program?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            int x = 3.14;
6.            int y = (int) Math.abs(x);
7.            System.out.print(y);
8.        }
9.    }
```

- a) 0**
- b) 3**
- c) 3.0**
- d) 3.1**

Answer: b

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
3
```

9. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              double x = 3.1;
6.              double y = 4.5;
7.              double z = Math.max( x, y );
8.              System.out.print(z);
9.          }
10.     }
```

a) true

b) flase

c) 3.1

d) 4.5

Answer: d

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
4.5
```

10. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              double x = 2.0;
6.              double y = 3.0;
7.              double z = Math.pow( x, y );
8.              System.out.print(z);
9.          }
10.     }
```

a) 2.0

- b) 4.0
- c) 8.0
- d) 9.0

Answer: c

Explanation: Math.pow(x, y) methods returns value of y to the power x, i.e x^y , $2.0^3.0 = 8.0$.

Output:

```
$ javac Output.java
$ java Output
8.0
```

1. Which of these exceptions is thrown by methods of System class?

- a) IOException
- b) SystemException
- c) SecurityException
- d) InputOutputException

Answer: c

Explanation: System class methods throw SecurityException.

2. Which of these methods initiates garbage collection?

- a) gc()
- b) garbage()
- c) garbagecollection()
- d) Systemgarbagecollection()

Answer: a

Explanation: None.

3. Which of these methods loads the specified dynamic library?

- a) load()
- b) library()
- c) loadlib()
- d) loadlibrary()

Answer: a

Explanation: load() methods loads the dynamic library whose name is specified.

4. Which of these method can set the out stream to OutputStream?

- a) setStream()
- b) setostream()
- c) setOut()
- d) streamtoOstream()

Answer: c

Explanation: None.

5. Which of these values are returns under the case of normal termination of a program?

- a) 0
- b) 1
- c) 2
- d) 3

Answer: a

Explanation: None.

6. What will be the output of the following Java program?


```

1.     import java.lang.System;
2.
3.     class Output
4.     {
5.
6.         public static void main(String args[])
7.         {
8.
9.             long start, end;
10.
11.            start = System.currentTimeMillis();
12.
13.            for (int i = 0; i < 10000000; i++);
14.
15.            end = System.currentTimeMillis();
16.
17.            System.out.print(end - start);
18.
19.        }
20.    }

```

- a) 0
- b) 1
- c) 1000
- d) System Dependent

Answer: d

Explanation: End time is the time taken by loop to execute it can be any non zero value depending on the System.

Output:

```

$ javac Output.java
$ java Output
78

```

7. What will be the output of the following Java program?

```

1.     import java.lang.System;
2.
3.     class Output
4.     {
5.
6.         public static void main(String args[])
7.         {
8.
9.             byte a[] = { 65, 66, 67, 68, 69, 70 };
10.
11.            byte b[] = { 71, 72, 73, 74, 75, 76 };
12.
13.            System.arraycopy(a, 0, b, 0, a.length);
14.
15.            System.out.print(new String(a) + " " + new String(b));
16.
17.        }
18.    }

```

- a) ABCDEF ABCDEF
- b) ABCDEF GHIJKL
- c) GHIJKL ABCDEF
- d) GHIJKL GHIJKL

Answer: a

Explanation: System.arraycopy() is a method of class System which is used to copy a string into another string.

Output:

```
$ javac Output.java
$ java Output
ABCDEF ABCDEF
```

8. What will be the output of the following Java program?

```
1.    import java.lang.System;
2.
3.    class Output
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            byte a[] = { 65, 66, 67, 68, 69, 70 };
10.
11.            byte b[] = { 71, 72, 73, 74, 75, 76 };
12.
13.            System.arraycopy(a, 0, b, 3, a.length - 3);
14.
15.            System.out.print(new String(a) + " " + new String(b));
16.
17.        }
18.    }
```

a) ABCDEF ABCDEF

b) ABCDEF GHIJKL

c) ABCDEF GHIABC

d) GHIJKL GHIJKL

Answer: c

Explanation: System.arraycopy() is a method of class System which is used to copy a string into another string.

Output:

```
$ javac Output.java
$ java Output
ABCDEF GHIABC
```

9. What will be the output of the following Java program?

```
1.    import java.lang.System;
2.
3.    class Output
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            byte a[] = { 65, 66, 67, 68, 69, 70 };
10.
11.            byte b[] = { 71, 72, 73, 74, 75, 76 };
12.
13.            System.arraycopy(a, 2, b, 3, a.length - 4);
14.
15.            System.out.print(new String(a) + " " + new String(b));
16.
17.        }
18.    }
```

10. }

11. }

- a) ABCDEF ABCDEF
- b) ABCDEF GHIJKL
- c) ABCDEF GHIABC
- d) ABCDEF GHICDL

Answer: d

Explanation: `System.arraycopy()` is a method of class `System` which is used to copy a string into another string.

Output:

```
$ javac Output.java
$ java Output
ABCDEF GHICDL
```

10. What will be the output of the following Java program?

```
1.      import java.lang.System;
2.
3.      class Output
4.      {
5.
6.          public static void main(String args[])
7.          {
8.              System.exit(5);
9.          }
10.     }
```

- a) 0
- b) 1
- c) 4
- d) 5

Answer: d

Explanation: None.

1. Which of these is a super class of wrappers Double and Float?

- a) Long
- b) Digits
- c) Float
- d) Number

Answer: d

Explanation: `Number` is an abstract class containing subclasses `Double`, `Float`, `Byte`, `Short`, `Integer` and `Long`.

2. Which of the following methods return the value as a double?

- a) `doubleValue()`
- b) `converDouble()`
- c) `getDouble()`
- d) `getDoubleValue()`

Answer: a

Explanation: None.

3. Which of these methods can be used to check whether the given value is a number or not?

- a) isNaN()
- b) isNumber()
- c) checkNaN()
- d) checkNumber()

Answer: a

Explanation: isNaN() methods returns true if num specified is not a number; otherwise it returns false.

4. Which of these method of Double wrapper can be used to check whether a given value is infinite or not?

- a) Infinite()
- b) isInfinite()
- c) checkInfinite()
- d) None of the mentioned

Answer: b

Explanation: isInfinite() methods returns true if the specified value is an infinite value otherwise it returns false.

5. Which of these exceptions is thrown by compareTo() method defined in a double wrapper?

- a) IOException
- b) SystemException
- c) CastException
- d) ClassCastException

Answer: d

Explanation: compareTo() methods compare the specified object to be double, if it is not then ClassCastException is thrown.

6. What will be the output of the following Java code?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              Double i = new Double(257.5);
6.              boolean x = i.isNaN();
7.              System.out.print(x);
8.          }
9.      }
```

- a) true
- b) false
- c) 0
- d) 1

Answer: b

Explanation: i.isNaN() method returns true if i is not a number and false when i is a number. Here false is returned because i is a number i.e 257.5.

Output:

```
$ javac Output.java
$ java Output
false
```

7. What will be the output of the following Java code?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            Double i = new Double(257.578);
6.            int x = i.intValue();
7.            System.out.print(x);
8.        }
9.    }
```

- a) 0
- b) 1
- c) 256
- d) 257

Answer: d

Explanation: *intValue()* method returns the value of wrapper *i* as a *Integer*. *i* is 257.578 is double number when converted to an integer data type its value is 257.

Output:

```
$ javac Output.java
$ java Output
257
```

8. What will be the output of the following Java code?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            Double i = new Double(257.578123456789);
6.            float x = i.floatValue();
7.            System.out.print(x);
8.        }
9.    }
```

- a) 0
- b) 257.0
- c) 257.57812
- d) 257.578123456789

Answer: c

Explanation: *floatValue()* converts the value of wrapper *i* into *float*, since *float* can measure till 5 places after decimal hence 257.57812 is stored in floating point variable *x*.

Output:

```
$ javac Output.java
$ java Output
257.57812
```

9. What will be the output of the following Java code?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              Double y = new Double(257.57812);
6.              Double i = new Double(257.578123456789);
7.              try
8.              {
9.                  int x = i.compareTo(y);
10.                 System.out.print(x);
11.             }
12.             catch(ClassCastException e)
13.             {
14.                 System.out.print("Exception");
15.             }
16.         }
17.     }
```

- a) 0
- b) 1
- c) Exception
- d) None of the mentioned

Answer: b

Explanation: i.compareTo() methods two double values, if they are equal then 0 is returned and if not equal then 1 is returned, here 257.57812 and 257.578123456789 are not equal hence 1 is returned and stored in x.

Output:

```
$ javac Output.java
$ java Output
1
```

1. Which of these packages contain classes and interfaces used for input & output operations of a program?

- a) java.util
- b) java.lang
- c) java.io
- d) all of the mentioned

Answer: c

Explanation: java.io provides support for input and output operations.

2. Which of these class is not a member class of java.io package?

- a) String
- b) StringReader
- c) Writer
- d) File

Answer: a

Explanation: None.

3. Which of these interface is not a member of java.io package?

- a) DataInput
- b) ObjectInput
- c) ObjectFilter
- d) FileFilter

Answer: c

Explanation: None.

4. Which of these class is not related to input and output stream in terms of functioning?

- a) File
- b) Writer
- c) InputStream
- d) Reader

Answer: a

Explanation: A File describes properties of a file, a File object is used to obtain or manipulate the information associated with a disk file, such as the permissions, time date, and directories path, and to navigate subdirectories.

5. Which of these is specified by a File object?

- a) a file in disk
- b) directory path
- c) directory in disk
- d) none of the mentioned

Answer: c

Explanation: None.

6. Which of these is method for testing whether the specified element is a file or a directory?

- a) IsFile()
- b) isFile()
- c) Isfile()
- d) isfile()

Answer: b

Explanation: isFile() returns true if called on a file and returns false when called on a directory.

7. What will be the output of the following Java code?

```
1.    import java.io.*;
2.
3.    class files
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            File obj = new File("/java/system");
10.
11.            System.out.print(obj.getName());
12.        }
13.    }
```

```
8.      }
9.    }
```

- a) java
- b) system
- c) java/system
- d) /java/system

Answer: b

Explanation: obj.getName() returns the name of the file.

Output:

```
$ javac files.java
$ java files
system
```

8. What will be the output of the following Java program? (Note: file is made in c drive.)

```
1.    import java.io.*;
2.
3.    class files
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            File obj = new File("/java/system");
10.           System.out.print(obj.getAbsolutePath());
11.
12.        }
13.    }
```

- a) java
- b) system
- c) java/system
- d) \java\system

Answer: d

Explanation: None.

Output:

```
$ javac files.java
$ java files
\java\system
```

9. What will be the output of the following Java program? (Note: file is made in c drive.)

```
1.    import java.io.*;
2.
3.    class files
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            File obj = new File("/java/system");
```



```

7.         System.out.print(obj.canWrite());
8.         System.out.print(" " + obj.canRead());
9.     }
10. }

```

- a) true false
- b) false true
- c) true true
- d) false false

Answer: d

Explanation: None.

Output:

```

$ javac files.java
$ java files
false false

```

10. What will be the output of the following Java program? (Note: file is made in c drive.)

```

1.     import java.io.*;
2.
3.     class files
4.     {
5.
6.         public static void main(String args[])
7.         {
8.
9.             File obj = new File("/java/system");
10.            System.out.print(obj.getParent());
11.            System.out.print(" " + obj.isFile());
12.        }
13.    }

```

- a) java true
- b) java false
- c) \java false
- d) \java true

Answer: c

Explanation: getParent() gives the parent directory of the file and isFile() checks whether the present file is a directory or a file in the disk.

Output:

```

$ javac files.java
$ java files
\java false

```

1. Which of these classes is used for input and output operation when working with bytes?

- a) InputStream
- b) Reader
- c) Writer
- d) All of the mentioned

Answer: a
Explanation: InputStream & OutputStream are designed for byte stream. Reader and writer are designed for character stream.

2. Which of these class is used to read and write bytes in a file?

- a) FileReader**
- b) FileWriter**
- c) FileInputStream**
- d) InputStreamReader**

Answer: c
Explanation: None.

3. Which of these method of InputStream is used to read integer representation of next available byte input?

- a) read()**
- b) scanf()**
- c) get()**
- d) getInteger()**

Answer: a
Explanation: None.

4. Which of these data type is returned by every method of OutputStream?

- a) int**
- b) float**
- c) byte**
- d) none of the mentioned**

Answer: d
Explanation: Every method of OutputStream returns void and throws an IOException in case of errors.

5. Which of these is a method to clear all the data present in output buffers?

- a) clear()**
- b) flush()**
- c) fflush()**
- d) close()**

Answer: b
Explanation: None.

6. Which of these method(s) is/are used for writing bytes to an outputstream?

- a) put()**
- b) print() and write()**
- c) printf()**
- d) write() and read()**

Answer: b
Explanation: write() and print() are the two methods of OutputStream that are used for printing the byte data.

7. What will be the output of the following Java program? (Note: inputoutput.java is stored in the disk.)

```
1.    import java.io.*;
2.
3.    class filesinputoutput
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.        }
```

```

5.      {
6.          InputStream obj = new FileInputStream("inputoutput.java");
7.          System.out.print(obj.available());
8.      }
9.  }

```

- a) true
- b) false
- c) prints number of bytes in file
- d) prints number of characters in the file

Answer: c

Explanation: obj.available() returns the number of bytes.

Output:

```

$ javac filesinputoutput.java
$ java filesinputoutput
1422

```

(Output will be different in your case)

8. What will be the output of the following Java program?

```

1.  import java.io.*;
2.  public class filesinputoutput
3.  {
4.      public static void main(String[] args)
5.      {
6.          String obj  = "abc";
7.          byte b[] = obj.getBytes();
8.          ByteArrayInputStream obj1 = new ByteArrayInputStream(b);
9.          for (int i = 0; i < 2; ++ i)
10.         {
11.             int c;
12.             while ((c = obj1.read()) != -1)
13.             {
14.                 if(i == 0)
15.                 {
16.                     System.out.print((char)c);
17.                 }
18.             }
19.         }

```

20. }

21. }

- a) abc
- b) ABC
- c) ab
- d) AB

Answer: a

Explanation: None.

Output:

```
$ javac filesinputoutput.java
$ java filesinputoutput
abc
```

9. What will be the output of the following Java program?

```
1.      import java.io.*;
2.
3.      public class filesinputoutput
4.      {
5.
6.          public static void main(String[] args)
7.          {
8.
9.              String obj  = "abc";
10.
11.              byte b[] = obj.getBytes();
12.
13.              ByteArrayInputStream obj1 = new ByteArrayInputStream(b);
14.
15.              for (int i = 0; i < 2; ++ i)
16.              {
17.
18.                  int c;
19.
20.                  while ((c = obj1.read()) != -1)
21.                  {
22.
23.                      if (i == 0)
24.                      {
25.
26.                          System.out.print(Character.toUpperCase((char)c));
27.
28.                      }
29.
30.                  }
31.
32.              }
33.
34.          }
35.
36.      }
```

- a) abc
- b) ABC
- c) ab
- d) AB

Answer: b

Explanation: None.

Output:

```
$ javac filesinputoutput.java
$ java filesinputoutput
ABC
```

10. What will be the output of the following Java program?

```
1.      import java.io.*;
2.
3.      public class filesinputoutput
4.      {
5.
6.          public static void main(String[] args)
7.          {
8.
9.              String obj  = "abc";
10.             byte b[] = obj.getBytes();
11.             ByteArrayInputStream obj1 = new ByteArrayInputStream(b);
12.             for (int i = 0; i < 2; ++ i)
13.             {
14.
15.                 int c;
16.                 while ((c = obj1.read()) != -1)
17.                 {
18.
19.                     if (i == 0)
20.                     {
21.                         System.out.print(Character.toUpperCase((char) c));
22.                         obj2.write(1);
23.                     }
24.                 }
25.                 System.out.print(obj2);
26.             }
27.         }
28.     }
```

a) AaBaCa

b) ABCaaa

c) AaaBaaCaa

d) AaBaaCaaa

Answer: d

Explanation: None.

Output:

```
$ javac filesinputoutput.java
$ java filesinputoutput
AaBaaCaaa
```

1. Which of these stream contains the classes which can work on character stream?

- a) InputStream**
- b) OutputStream**
- c) Character Stream**
- d) All of the mentioned**

Answer: c

Explanation: InputStream & OutputStream classes under byte stream they are not streams. Character Stream contains all the classes which can work with Unicode.

2. Which of these class is used to read characters in a file?

- a) FileReader**
- b) FileWriter**
- c) FileInputStream**
- d) InputStreamReader**

Answer: a

Explanation: None.

3. Which of these method of FileReader class is used to read characters from a file?

- a) read()**
- b) scanf()**
- c) get()**
- d) getInteger()**

Answer: a

Explanation: None.

4. Which of these class can be used to implement the input stream that uses a character array as the source?

- a) BufferedReader**
- b) FileReader**
- c) CharArrayReader**
- d) FileArrayReader**

Answer: c

Explanation: CharArrayReader is an implementation of an input stream that uses character array as a source. Here array is the input source.

5. Which of these classes can return more than one character to be returned to input stream?

- a) BufferedReader**
- b) Bufferedwriter**
- c) PushbachReader**
- d) CharArrayReader**

Answer: c

Explanation: PushbackReader class allows one or more characters to be returned to the input stream. This allows looking ahead in input stream and performing action accordingly.

6. What will be the output of the following Java program?

```
1.      import java.io.*;
2.
3.      class Chararrayinput
4.      {
```

```

4.         public static void main(String[] args)
5.         {
6.             String obj  = "abcdef";
7.             int length = obj.length();
8.             char c[] = new char[length];
9.             obj.getChars(0,length,c,0);
10.            CharArrayReader input1 = new CharArrayReader(c);
11.            CharArrayReader input2 = new CharArrayReader(c, 0, 3);
12.            int i;
13.            try
14.            {
15.                while ((i = input1.read()) != -1)
16.                {
17.                    System.out.print((char)i);
18.                }
19.            }
20.            catch (IOException e)
21.            {
22.                // TODO Auto-generated catch block
23.                e.printStackTrace();
24.            }
25.        }
26.    }

```

- a) abc
- b) abcd
- c) abcde
- d) abcdef

Answer: d

Explanation: None.

Output:

```

$ javac Chararrayinput.java
$ java Chararrayinput
abcdef

```

7. What will be the output of the following Java program?

```

1.     import java.io.*;
2.

```

```

3.      {
4.          public static void main(String[] args)
5.          {
6.              String obj  = "abcdef";
7.              int length = obj.length();
8.              char c[] = new char[length];
9.              obj.getChars(0, length, c, 0);
10.             CharArrayReader input1 = new CharArrayReader(c);
11.             CharArrayReader input2 = new CharArrayReader(c, 0, 3);
12.             int i;
13.             try
14.             {
15.                 while ((i = input2.read()) != -1)
16.                 {
17.                     System.out.print((char)i);
18.                 }
19.             }
20.             catch (IOException e)
21.             {
22.                 // TODO Auto-generated catch block
23.                 e.printStackTrace();
24.             }
25.         }
26.     }

```

- a) abc
- b) abcd
- c) abcde
- d) abcdef

Answer: a

Explanation: None.

Output:

```

$ javac Chararrayinput.java
$ java Chararrayinput
abc

```

8. What will be the output of the following Java program?

```

1.      import java.io.*;

```



```

2.     class Chararrayinput
3.     {
4.         public static void main(String[] args)
5.         {
6.             String obj  = "abcdefgh";
7.             int length = obj.length();
8.             char c[] = new char[length];
9.             obj.getChars(0, length, c, 0);
10.            CharArrayReader input1 = new CharArrayReader(c);
11.            CharArrayReader input2 = new CharArrayReader(c, 1, 4);
12.            int i;
13.            int j;
14.            try
15.            {
16.                while ((i = input1.read()) == (j = input2.read()))
17.                {
18.                    System.out.print((char)i);
19.                }
20.            }
21.            catch (IOException e)
22.            {
23.                // TODO Auto-generated catch block
24.                e.printStackTrace();
25.            }
26.        }
27.    }

```

- a) abc
- b) abcd
- c) abcde
- d) none of the mentioned

Answer: d

Explanation: No output is printed. CharArrayReader object input1 contains string “abcdefgh” whereas object input2 contains string “bcde”, when while((i=input1.read())==(j=input2.read())) is executed the starting character of each object is compared since they are unequal control comes out of loop and nothing is printed on the screen.

Output:

```

$ javac Chararrayinput.java
$ java Chararrayinput

```

1. Which of the following is not a segment of memory in java?

- a) Stack Segment**
- b) Heap Segment**
- c) Code Segment**
- d) Register Segment**

Answer: d

Explanation: There are only 3 types of memory segment. Stack Segment, Heap Segment and Code Segment.
Answer: a
Explanation: Code Segment loads compiled java bytecode. Bytecode is platform independent.

3. What is JVM?

- a) Bootstrap**
- b) Interpreter**
- c) Extension**
- d) Compiler**

Answer: b

Explanation: JVM is Interpreter. It reads .class files which is the byte code generated by compiler line by line and converts it into native OS code.

4. Which one of the following is a class loader?

- a) Bootstrap**
- b) Compiler**
- c) Heap**
- d) Interpreter**

Answer: a

Explanation: Bootstrap is a class loader. It loads the classes into memory.

5. Which class loader loads jar files from JDK directory?

- a) Bootstrap**
- b) Extension**
- c) System**
- d) Heap**

Answer: b

Explanation: Extension loads jar files from lib/ext directory of the JRE. This gives the basic functionality available.

6. Which of the following is not a memory classification in java?

- a) Young**
- b) Old**
- c) Permanent**
- d) Temporary**

Answer: d

Explanation: Young generation is further classified into Eden space and Survivor space. Old generation is also the tenured space. The permanent generation is the non heap space.

7. What is the Java 8 update of PermGen?

- a) Code Cache**
- b) Tenured Space**
- c) Metaspace**
- d) Eden space**

Answer: c

Explanation: Metaspace is the replacement of PermGen in Java 8. It is very similar to PermGen except that it resizes itself dynamically. Thus, it is unbounded.

8. Classes and Methods are stored in which space?

- a) Eden space
- b) Survivor space
- c) Tenured space
- d) Permanent space

Answer: d

Explanation: The permanent generation holds objects which JVM finds convenient to have the garbage collector. Objects describing classes and methods, as well as the classes and methods themselves, are a part of Permanent generation.

9. Where is String Pool stored?

- a) Java Stack
- b) Java Heap
- c) Permanent Generation
- d) Metaspace

Answer: b

Explanation: When a string is created; if the string already exists in the pool, the reference of the existing string will be returned, else a new object is created and its reference is returned.

Answer: a
Explanation: We can import the same package or same class multiple times. Neither compiler nor JVM complains will complain about it. JVM will internally load the class only once no matter how many times we import the same class or package.

1. Which of these exceptions handles the situations when an illegal argument is used to invoke a method?

- a) `IllegalArgumentException`
- b) `Argument Exception`
- c) `IllegalArgumentExpection`
- d) `IllegalMethodArgumentExcepetion`

Answer: c

Explanation: None.

2. Which of these exceptions will be thrown if we declare an array with negative size?

- a) `IllegalArrayException`
- b) `IllegalArraySizeExeption`
- c) `NegativeArrayException`
- d) `NegativeArraySizeExeption`

Answer: d

Explanation: Array size must always be positive if we declare an array with negative size then built in exception “`NegativeArraySizeException`” is thrown by the java’s run time system.

3. Which of these packages contain all the Java’s built in exceptions?

- a) `java.io`
- b) `java.util`
- c) `java.lang`
- d) `java.net`

Answer: c

Explanation: None.

4. Which of these exceptions will be thrown if we use null reference for an arithmetic operation?

- a) `ArithmeticException`
- b) `NullPointerException`
- c) `IllegalAccessExpection`
- d) `IllegalOperationException`

Answer: b

Explanation: If we use null reference anywhere in the code where the value stored in that reference is used then *NullPointerException* occurs.

5. Which of these class is used to create user defined exception?

- a) java.lang
- b) Exception
- c) RunTime
- d) System

Answer: b

Explanation: Exception class contains all the methods necessary for defining an exception. The class contains the *Throwable* class.

6. What will be the output of the following Java program?

```
1.      class exception_handling
2.      {
3.          public static void main(String args[])
4.          {
5.              try
6.              {
7.                  int a[] = {1, 2,3 , 4, 5};
8.                  for (int i = 0; i < 7; ++i)
9.                      System.out.print(a[i]);
10.             }
11.             catch (ArrayIndexOutOfBoundsException e)
12.             {
13.                 System.out.print("0");
14.             }
15.         }
16.     }
```

- a) 12345
- b) 123450
- c) 1234500
- d) Compilation Error

Answer: b

Explanation: When array index goes out of bound then *ArrayIndexOutOfBoundsException* exception is thrown by the system.

Output:

```
$ javac exception_handling.java
$ java exception_handling
123450
```

7. What will be the output of the following Java program?

```

1.     class exception_handling
2.     {
3.         public static void main(String args[])
4.         {
5.             try
6.             {
7.                 int a[] = {1, 2,3 , 4, 5};
8.                 for (int i = 0; i < 5; ++i)
9.                     System.out.print(a[i]);
10.                int x = 1/0;
11.            }
12.            catch (ArrayIndexOutOfBoundsException e)
13.            {
14.                System.out.print("A");
15.            }
16.            catch (ArithmeticException e)
17.            {
18.                System.out.print("B");
19.            }
20.        }
21.    }

```

- a) 12345
- b) 12345A
- c) 12345B
- d) Compilation Error

Answer: c

Explanation: There can be more than one catch of a single try block. Here Arithmetic exception occurs instead of Array index out of bound exception hence B is printed after 12345

Output:

```

$ javac exception_handling.java
$ java exception_handling
12345B

```

8. What will be the output of the following Java program?

```

1.     class exception_handling
2.     {
3.         static void throwexception() throws ArithmeticException
4.         {

```

```

5.         System.out.print("0");
6.         throw new ArithmeticException ("Exception");
7.     }
8.     public static void main(String args[])
9.     {
10.        try
11.        {
12.            throwexception();
13.        }
14.        catch (ArithmeticException e)
15.        {
16.            System.out.println("A");
17.        }
18.    }
19. }

```

- a) A
- b) 0
- c) 0A
- d) Exception

Answer: c

Explanation: None.

Output:

```

$ javac exception_handling.java
$ java exception_handling
0A

```

9. What will be the output of the following Java program?

```

1.     class exception_handling
2.     {
3.         public static void main(String args[])
4.         {
5.             try
6.             {
7.                 int a = 1;
8.                 int b = 10 / a;
9.             }
10.        }

```

```

11.             if (a == 1)
12.
13.                 a = a / a - a;
14.
15.             if (a == 2)
16.
17.                 {
18.
19.                     int c[] = {1};
20.
21.                     c[8] = 9;
22.
23.                 }
24.
25.             finally
26.
27.             {
28.
29.                 System.out.print("A");
30.
31.             }
32.
33.         }
34.
35.     catch (NullPointerException e)
36.
37.     {
38.
39.         System.out.println("B");
40.
41.     }
42.
43. }

```

- a) A
- b) B
- c) AB
- d) BA

Answer: a

Explanation: The inner try block does not have a catch which can tackle `ArrayIndexOutOfBoundsException` hence finally is executed which prints 'A' the outer try block does have catch for `NullPointerException` exception but no such exception occurs in it hence its catch is never executed and only 'A' is printed.

Output:

```

$ javac exception_handling.java
$ java exception_handling
A

```

10. What will be the output of the following Java program?

```

1.     class exception_handling
2.     {
3.
4.         public static void main(String args[])
5.
6.         {
7.
8.             try
9.
10.            {
11.
12.            }
13.
14.        }
15.    }

```

```

6.      {
7.          int a = args.length;
8.          int b = 10 / a;
9.          System.out.print(a);
10.     try
11.     {
12.         if (a == 1)
13.             a = a / a - a;
14.         if (a == 2)
15.         {
16.             int c = {1};
17.             c[8] = 9;
18.         }
19.     }
20.     catch (ArrayIndexOutOfBoundsException e)
21.     {
22.         System.out.println("TypeA");
23.     }
24.     catch (ArithmeticException e)
25.     {
26.         System.out.println("TypeB");
27.     }
28. }
29. }

```

Note: Execution command line: \$ java exception_handling one two

a) TypeA

b) TypeB

c) 0TypeA

d) 0TypeB

Answer: d

Explanation: Execution command line is “\$ java exception_handling one two” hence there are two input making args.length = 2, hence “c[8] = 9” in second try block is executing which throws ArrayIndexOutOfBoundsException which is caught by catch of nested try block. Hence 0TypeB is printed.

Output:

```

$ javac exception_handling.java
$ java exception_handling
0TypeB

```

1. Which of these class provides various types of rounding functions?

- a) Math
- b) Process
- c) System
- d) Object

Answer: a

Explanation: None.

2. Which of these methods return a smallest whole number greater than or equal to variable X?

- a) double ceil(double X)
- b) double floor(double X)
- c) double max(double X)
- d) double min(double X)

Answer: a

Explanation: ceil(double X) returns the smallest whole number greater than or equal to variable X.

3. Which of these method returns a largest whole number less than or equal to variable X?

- a) double ceil(double X)
- b) double floor(double X)
- c) double max(double X)
- d) double min(double X)

Answer: b

Explanation: double floor(double X) returns a largest whole number less than or equal to variable X.

4. Which of function return absolute value of a variable?

- a) abs()
- b) absolute()
- c) absolutevariable()
- d) none of the mentioned

Answer: a

Explanation: abs() returns the absolute value of a variable.

5. What will be the output of the following Java code?

```
1.      class A
2.      {
3.          int x;
4.          int y;
5.          void display()
6.          {
7.              System.out.print(x + " " + y);
8.          }
9.      }
10.     class Output
11.     {
12.         public static void main(String args[])
13.         {
```

```

14.         A obj1 = new A();
15.         A obj2 = new A();
16.         obj1.x = 1;
17.         obj1.y = 2;
18.         obj2 = obj1.clone();
19.         obj1.display();
20.         obj2.display();
21.     }
22. }

```

- a) 1 2 0 0
- b) 1 2 1 2
- c) 0 0 0 0
- d) System Dependent

Answer: b
Explanation: clone() method of object class is used to generate duplicate copy of the object on which it is called. Copy of obj1 is generated and stored in obj2.
Output:

```

$ javac Output.java
$ java Output
1 2 1 2

```

6. What will be the output of the following Java code?

```

1.     class Output
2.     {
3.         public static void main(String args[])
4.         {
5.             double x = 3.14;
6.             int y = (int) Math.abs(x);
7.             System.out.print(y);
8.         }
9.     }

```

- a) 0
- b) 3
- c) 3.0
- d) 3.1

Answer: b
Explanation: None.
Output:

```

$ javac Output.java
$ java Output
3

```

7. What will be the output of the following Java code?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            double x = 3.14;
6.            int y = (int) Math.ceil(x);
7.            System.out.print(y);
8.        }
9.    }
```

- a) 0
- b) 3
- c) 3.0
- d) 4

Answer: d

Explanation: *ciel(double X)* returns the smallest whole number greater than or equal to variable x.

Output:

```
$ javac Output.java
$ java Output
4
```

8. What will be the output of the following Java code?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            double x = 3.14;
6.            int y = (int) Math.floor(x);
7.            System.out.print(y);
8.        }
9.    }
```

- a) 0
- b) 3
- c) 3.0
- d) 4

Answer: b

Explanation: *double floor(double X)* returns a largest whole number less than or equal to variable X. Here the smallest whole number less than 3.14 is 3.

Output:

```
$ javac Output.java
```

1. Which of these methods of Byte wrapper can be used to obtain Byte object from a string?

- a) toString()
- b) getString()
- c) decode()
- d) encode()

Answer: c

Explanation: decode() methods returns a Byte object that contains the value specified by string.

2. Which of the following methods Byte wrapper return the value as a double?

- a) doubleValue()
- b) converDouble()
- c) getDouble()
- d) getDoubleValue()

Answer: a

Explanation: doubleValue() returns the value of invoking object as double.

3. Which of these is a super class of wrappers Byte and short wrappers?

- a) Long
- b) Digits
- c) Float
- d) Number

Answer: d

Explanation: Number is an abstract class containing subclasses Double, Float, Byte, Short, Integer and Long.

4. Which of these methods is not defined in both Byte and Short wrappers?

- a) intValue()
- b) isInfinite()
- c) toString()
- d) hashCode()

Answer: b

Explanation: isInfinite() methods is defined in Integer and Long Wrappers, returns true if specified value is an infinite value otherwise it returns false.

5. What will be the output of the following Java code?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              Double i = new Double(257.5);
6.              Double x = i.MAX_VALUE;
7.              System.out.print(x);
8.          }
9.      }
```

a) 0

- b) 1.7976931348623157E308
- c) 1.7976931348623157E30
- d) None of the mentioned

Answer: b

Explanation: The super class of Double class defines a constant MAX_VALUE above which a number is considered to be infinity. MAX_VALUE is 1.7976931348623157E308.

Output:

```
$ javac Output.java
$ java Output
1.7976931348623157E308
```

6. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              Double i = new Double(257.5);
6.              Double x = i.MIN_VALUE;
7.              System.out.print(x);
8.          }
9.      }
```

- a) 0
- b) 4.9E-324
- c) 1.7976931348623157E308
- d) None of the mentioned

Answer: b

Explanation: The super class of Byte class defines a constant MIN_VALUE below which a number is considered to be negative infinity. MIN_VALUE is 4.9E-324.

Output:

```
$ javac Output.java
$ java Output
4.9E-324
```

7. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              Double i = new Double(257.578123456789);
6.              float x = i.floatValue();
7.              System.out.print(x);
8.          }
```

9. }

- a) 0
- b) 257.0
- c) 257.57812
- d) 257.578123456789

Answer: c

Explanation: floatValue() converts the value of wrapper i into float, since float can measure till 5 places after decimal hence 257.57812 is stored in floating point variable x.

Output:

```
$ javac Output.java
$ java Output
257.57812
```

1. Which of these methods of Character wrapper can be used to obtain the char value contained in Character object.

- a) get()
- b) getVhar()
- c) charValue()
- d) getCharacter()

Answer: c

Explanation: To obtain the char value contained in a Character object, we use charValue() method.

2. Which of the following constant are defined in Character wrapper?

- a) MAX_RADIX
- b) MAX_VALUE
- c) TYPE
- d) All of the mentioned

Answer: d

Explanation: Character wrapper defines 5 constants – MAX_RADIX, MIN_RADIX, MAX_VALUE, MIN_VALUE & TYPE.

3. Which of these is a super class of Character wrapper?

- a) Long
- b) Digits
- c) Float
- d) Number

Answer: d

Explanation: Number is an abstract class containing subclasses Double, Float, Byte, Short, Character, Integer and Long.

4. Which of these methods is used to know whether a given Character object is part of Java's Identifiers?

- a) isIdentifier()
- b) isJavaIdentifier()
- c) isJavaIdentifierPart()
- d) none of the mentioned

Answer: c

Explanation: None.

5. Which of these coding techniques is used by method isDefined()?

- a) Latin
- b) ASCII
- c) ANSI
- d) UNICODE

Answer: d

Explanation: isDefined() returns true if ch is defined by Unicode. Otherwise, it returns false.

6. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              int a = Character.MAX_VALUE;
6.              System.out.print((char)a);
7.          }
8.      }
```

a) <

b) >

c) ?

d) \$

Answer: c

Explanation: Character.MAX_VALUE returns the largest character value, which is of character '?'.

Output:

```
$ javac Output.java
$ java Output
?
```

7. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              int a = Character.MIN_VALUE;
6.              System.out.print((char)a);
7.          }
8.      }
```

a) <

b) !

c) @

d) Space

Answer: d

Explanation: Character.MIN_VALUE returns the smallest character value, which is of space character ' '.

Output:

```
$ javac Output.java
$ java Output
```

8. What will be the output of the following Java program?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            char a[] = {'a', '5', 'A', ' '};
6.            System.out.print(Character.isDigit(a[0])+ " ");
7.            System.out.print(Character.isWhitespace(a[3])+ " ");
8.            System.out.print(Character.isUpperCase(a[2]));
9.        }
10.   }
```

- a) true false true
- b) false true true
- c) true true false
- d) false false false

Answer: b

Explanation: `Character.isDigit(a[0])` checks for `a[0]`, whether it is a digit or not, since `a[0]` i.e 'a' is a character `false` is returned. `a[3]` is a whitespace hence `Character.isWhitespace(a[3])` returns a `true`. `a[2]` is an uppercase letter i.e 'A' hence `Character.isUpperCase(a[2])` returns `true`.

Output:

```
$ javac Output.java
$ java Output
false true true
```

9. What will be the output of the following Java program?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            char a = (char) 98;
6.            a = Character.toUpperCase(a);
7.            System.out.print(a);
8.        }
9.    }
```

- a) b
- b) c
- c) B
- d) C

Answer: c

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
B
```

10. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              char a = '@';
6.              boolean x = Character.isLetter(a);
7.              System.out.print(x);
8.          }
9.      }
```

- a) true
- b) false
- c) @
- d) B

Answer: b

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
false
```

1. Which of these methods of Boolean wrapper returns boolean equivalent of an object.

- a) getBool()
- b) booleanValue()
- c) getbooleanValue()
- d) getboolValue()

Answer: b

Explanation: None.

2. Which of the following constant are defined in Boolean wrapper?

- a) TRUE
- b) FALSE
- c) TYPE
- d) All of the mentioned

Answer: d

Explanation: Boolean wrapper defines 3 constants – TRUE, FALSE & TYPE.

3. Which of these methods return string equivalent of Boolean object?

- a) getString()
- b) toString()
- c) converString()
- d) getStringObject()

Answer: b
Explanation: None.

4. Which of these methods is used to know whether a string contains “true”?
- a) valueOf()
 - b) valueOfString()
 - c) getString()
 - d) none of the mentioned

Answer: a
Explanation: valueOf() returns true if the specified string contains “true” in lower or uppercase and false otherwise.

5. Which of these class have only one field?
- a) Character
 - b) Boolean
 - c) Byte
 - d) void

Answer: d
Explanation: Void class has only one field – TYPE, which holds a reference to the Class object for type void. We do not create an instance of this class.

6. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              String str = "true";
6.              boolean x = Boolean.valueOf(str);
7.              System.out.print(x);
8.          }
9.      }
```

- a) True
- b) False
- c) Compilation Error
- d) Runtime Error

Answer: a
Explanation: valueOf() returns true if the specified string contains “true” in lower or uppercase and false otherwise.
Output:

```
$ javac Output.java
$ java Output
true
```

7. What will be the output of the following Java program?

```
1.      class Output
2.      {
```

```

3.         public static void main(String args[])
4.         {
5.             String str = "true false true";
6.             boolean x = Boolean.valueOf(str);
7.             System.out.print(x);
8.         }
9.     }

```

- a) True
- b) False
- c) Compilation Error
- d) Runtime Error

Answer: b

Explanation: valueOf() returns true if the specified string contains “true” in lower or uppercase and false otherwise.

Output:

```

$ javac Output.java
$ java Output
false

```

8. What will be the output of the following Java program?

```

1.     class Output
2.     {
3.         public static void main(String args[])
4.         {
5.             String str = "TRUE";
6.             boolean x = Boolean.valueOf(str);
7.             System.out.print(x);
8.         }
9.     }

```

- a) True
- b) False
- c) Compilation Error
- d) Runtime Error

Answer: a

Explanation: valueOf() returns a Boolean instance representing the specified boolean value. If the specified boolean value is true, this method returns Boolean.TRUE; if it is false, this method returns Boolean.FALSE. If a new Boolean instance is not required, this method should generally be used in preference to the constructor Boolean(boolean), as this method is likely to yield significantly better space and time.

Output:

```

$ javac Output.java
$ java Output
true

```

9. What will be the output of the following Java program?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            String str = "true false";
6.            boolean x = Boolean.parseBoolean(str);
7.            System.out.print(x);
8.        }
9.    }
```

- a) True
- b) False
- c) System Dependent
- d) Compilation Error

Answer: b

Explanation: `parseBoolean()` Parses the string argument as a boolean. The boolean returned represents the value true if the string argument is not null and is equal, ignoring case, to the string “true”.

Example: `Boolean.parseBoolean(“True”)` returns true.

Example: `Boolean.parseBoolean(“yes”)` returns false.

Output:

```
$ javac Output.java
$ java Output
false
```

10. What will be the output of the following Java program?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            String x = Boolean.toString(false);
6.        }
7.    }
```

- a) True
- b) False
- c) System Dependent
- d) Compilation Error

Answer: b

Explanation: `toString()` Returns a String object representing the specified boolean. If the specified boolean is true, then the string “true” will be returned, otherwise the string “false” will be returned.

Output:

```
$ javac Output.java
```

1. Which of these class contains all the methods present in Math class?

- a) SystemMath**
- b) StrictMath**
- c) Compiler**
- d) ClassLoader**

Answer: b

Explanation: SystemMath class defines a complete set of mathematical methods that are parallel those in Math class. The difference is that the StrictMath version is guaranteed to generate precisely identical results across all Java implementations.

2. Which of these method return a pseudorandom number?

- a) rand()**
- b) random()**
- c) randomNumber()**
- d) randGenerator()**

Answer: b

Explanation: None.

3. Which of these method returns the remainder of dividend / divisor?

- a) remainder()**
- b) getRemainder()**
- c) CSRemainder()**
- d) IEEEremainder()**

Answer: d

Explanation: IEEEremainder() returns the remainder of dividend / divisor.

4. Which of these method converts radians to degrees?

- a) toRadian()**
- b) toDegree()**
- c) convertRadian()**
- d) converDegree()**

Answer: b

Explanation: None.

5. toRadian() and toDegree() methods were added by which version of Java?

- a) Java 1.0**
- b) Java 1.5**
- c) Java 2.0**
- d) Java 3.0**

Answer: c

Explanation: toRadian() and toDegree() methods were added by Java 2.0 before that there was no method which could directly convert degree into radians and vice versa.

6. Which of these method returns a smallest whole number greater than or equal to variable X?

- a) double ciel(double X)**
- b) double floor(double X)**
- c) double max(double X)**
- d) double min(double X)**

Answer: a

Explanation: ceil(double X) returns the smallest whole number greater than or equal to variable X.

7. What will be the output of the following Java program?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            double x = 3.14;
6.            int y = (int) Math.toDegrees(x);
7.            System.out.print(y);
8.        }
9.    }
```

- a) 0
- b) 179
- c) 180
- d) 360

Answer: b

Explanation: 3.14 in degree 179.9087. We usually take it to be 180. Buts here we have type casted it to integer data type hence 179.

Output:

```
$ javac Output.java
$ java Output
179
```

8. What will be the output of the following Java program?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            double x = 3.14;
6.            int y = (int) Math.toRadians(x);
7.            System.out.print(y);
8.        }
9.    }
```

- a) 0
- b) 3
- c) 3.0
- d) 3.1

Answer: a

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
0
```

9. What will be the output of the following Java program?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              double x = 102;
6.              double y = 5;
7.              double z = Math.IEEEremainder(x, y);
8.              System.out.print(z);
9.          }
10.     }
```

- a) 0
- b) 1
- c) 2
- d) 3

Answer: c

Explanation: IEEEremainder() returns the remainder of dividend / divisor. Here dividend is 102 and divisor is 5 therefore remainder is 2. It is similar to modulus – ‘%’ operator of C/C++ language.

Output:

```
$ javac Output.java
$ java Output
2
```

10. Will this Java program generate same output is executed again?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              int y = double z = Math.random();
6.              System.out.print(y);
7.          }
8.     }
```

- a) Yes
- b) No
- c) Compiler Dependent
- d) Operating System Dependent

Answer: b

Explanation: There is no relation between random numbers generated previously in Java.

1. Which of these classes encapsulate runtime environment?

- a) Class**
- b) System**
- c) Runtime**
- d) ClassLoader**

Answer: c

Explanation: None.

2. Which of the following exceptions is thrown by every method of Runtime class?

- a) IOException**
- b) SystemException**
- c) SecurityException**
- d) RuntimeException**

Answer: c

Explanation: Every method of Runtime class throws SecurityException.

3. Which of these methods returns the total number of bytes of memory available to the program?

- a) getMemory()**
- b) TotalMemory()**
- c) SystemMemory()**
- d) getProcessMemory()**

Answer: b

Explanation: TotalMemory() returns the total number of bytes available to the program.

4. Which of these Exceptions is thrown by loadClass() method of ClassLoader class?

- a) IOException**
- b) SystemException**
- c) ClassFormatError**
- d) ClassNotFoundException**

Answer: d

Explanation: None.

5. What will be the output of the following Java program?

```
1.      class X
2.      {
3.          int a;
4.          double b;
5.      }
6.      class Y extends X
7.      {
8.          int c;
9.      }
10.     class Output
```



```

11.      {
12.          public static void main(String args[])
13.      {
14.          X a = new X();
15.          Y b = new Y();
16.          Class obj;
17.          obj = b.getClass();
18.          System.out.print(obj.getSuperclass());
19.      }
20.  }

```

- a) X
- b) Y
- c) class X
- d) class Y

Answer: c

Explanation: getSuperClass() returns the super class of an object. b is an object of class Y which extends class X, Hence Super class of b is X. therefore class X is printed.

Output:

```

$ javac Output.java
$ java Output
class X

```

6. What will be the output of the following Java program?

```

1.      class X
2.      {
3.          int a;
4.          double b;
5.      }
6.      class Y extends X
7.      {
8.          int c;
9.      }
10.     class Output
11.     {
12.         public static void main(String args[])
13.     {
14.         X a = new X();
15.         Y b = new Y();

```

```
16.         Class obj;
17.         obj = b.getClass();
18.         System.out.print(b.equals(a));
19.     }
20. }
```

- a) 0
- b) 1
- c) true
- d) false

Answer: d

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
false
```

7. What will be the output of the following Java program?

```
1.     class X
2.     {
3.         int a;
4.         double b;
5.     }
6.     class Y extends X
7.     {
8.         int c;
9.     }
10.    class Output
11.    {
12.        public static void main(String args[])
13.        {
14.            X a = new X();
15.            Y b = new Y();
16.            Class obj;
17.            obj = b.getClass();
18.            System.out.print(obj.isInstance(a));
19.        }
20.    }
```

- a) 0
- b) 1
- c) true
- d) false

Answer: d

Explanation: Although class Y extends class X but still a is not considered related to Y hence `isInstance()` returns false.

Output:

```
$ javac Output.java
$ java Output
false
```

1. Which of these classes encapsulate runtime state of an object?

- a) Class
- b) System
- c) Runtime
- d) Cache

Answer: a

Explanation: None.

2. Which of these methods returns the class of an object?

- a) `getClass()`
- b) `Class()`
- c) `WhoseClass()`
- d) `WhoseObject()`

Answer: a

Explanation: None.

3. Which of these methods return a class object given its name?

- a) `getClass()`
- b) `findClass()`
- c) `getSystemClass()`
- d) `findSystemClass()`

Answer: d

Explanation: `findSystemClass()` returns a class object given its name.

4. Which of these class defines how the classes are loaded?

- a) Class
- b) System
- c) Runtime
- d) `ClassLoader`

Answer: d

Explanation: None.

5. What will be the output of the following Java program?

```
1.      class X
2.      {
3.          int a;
4.          double b;
5.      }
```

```

6.      class Y extends X
7.      {
8.          int c;
9.      }
10.     class Output
11.     {
12.         public static void main(String args[])
13.         {
14.             X a = new X();
15.             Y b = new Y();
16.             Class obj;
17.             obj = a.getClass();
18.             System.out.print(obj.getName());
19.         }
20.     }

```

- a) X
- b) Y
- c) a
- d) b

Answer: a

Explanation: getClass() is used to obtain the class of an object, here 'a' is an object of class 'X'. hence a.getClass() returns 'X' which is stored in class Class object obj.

Output:

```

$ javac Output.java
$ java Output
X

```

6. What will be the output of the following Java program?

```

1.      class X
2.      {
3.          int a;
4.          double b;
5.      }
6.      class Y extends X
7.      {
8.          int c;
9.      }
10.     class Output

```

```

11.      {
12.          public static void main(String args[])
13.      {
14.          X a = new X();
15.          Y b = new Y();
16.          Class obj;
17.          obj = b.getClass();
18.          System.out.print(obj.getSuperclass());
19.      }
20.  }

```

- a) X
- b) Y
- c) class X
- d) class Y

Answer: c

Explanation: getSuperClass() returns the super class of an object. b is an object of class Y which extends class X, Hence Super class of b is X. therefore class X is printed.

Output:

```

$ javac Output.java
$ java Output
class X

```

7. What will be the output of the following Java program?

```

1.      class X
2.      {
3.          int a;
4.          double b;
5.      }
6.      class Y extends X
7.      {
8.          int c;
9.      }
10.     class Output
11.     {
12.         public static void main(String args[])
13.     {
14.         X a = new X();
15.         Y b = new Y();

```

```
16.         Class obj;  
  
17.         obj = b.getClass();  
  
18.         System.out.print(obj.isLocalClass());  
  
19.     }  
  
20. }
```

- a) 0
- b) 1
- c) true
- d) false

Answer: d

Explanation: None.

Output:

```
$ javac Output.java  
$ java Output  
false
```

1. Which of the interface contains all the methods used for handling thread related operations in Java?

- a) Runnable interface
- b) Math interface
- c) System interface
- d) ThreadHandling interface

Answer: a

Explanation: Runnable interface defines all the methods for handling thread operations in Java.

2. Which of these class is used to make a thread?

- a) String
- b) System
- c) Thread
- d) Runnable

Answer: c

Explanation: 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.

3. Which of this interface is implemented by Thread class?

- a) Runnable
- b) Connections
- c) Set
- d) MapConnections

Answer: a

Explanation: None.

4. Which of these methods of a Thread class is used to suspend a thread for a period of time?

- a) sleep()
- b) terminate()
- c) suspend()
- d) stop()

Answer: a

Explanation: None.

5. What will be the output of the following Java program?

```
1.      class newthread implements Runnable
2.      {
3.          Thread t1,t2;
4.          newthread()
5.          {
6.              t1 = new Thread(this,"Thread_1");
7.              t2 = new Thread(this,"Thread_2");
8.              t1.start();
9.              t2.start();
10.         }
11.         public void run()
12.         {
13.             t2.setPriority(Thread.MAX_PRIORITY);
14.             System.out.print(t1.equals(t2));
15.         }
16.     }
17.     class multithreaded_programing
18.     {
19.         public static void main(String args[])
20.         {
21.             new newthread();
22.         }
23.     }
```

- a) true
- b) false
- c) true true
- d) false false

Answer: d

Explanation: Threads t1 & t2 are created by class newthread that is implementing runnable interface, hence both the threads are provided their own run() method specifying the actions to be taken. When constructor of newthread class is called first the run() method of t1 executes then the run method of t2 printing 2 times "false" as both the threads are not equal one is having different priority than other, hence falsefalse is printed.

Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
falsefalse
```

6. What will be the output of the following Java program?

```

1.     class newthread implements Runnable
2.     {
3.         Thread t;
4.         newthread()
5.         {
6.             t = new Thread(this,"New Thread");
7.             t.start();
8.         }
9.         public void run()
10.        {
11.            t.setPriority(Thread.MAX_PRIORITY);
12.            System.out.println(t);
13.        }
14.    }
15.    class multithreaded_programing
16.    {
17.        public static void main(String args[])
18.        {
19.            new newthread();
20.        }
21.    }

```

- a) Thread[New Thread,0,main]
- b) Thread[New Thread,1,main]
- c) Thread[New Thread,5,main]
- d) Thread[New Thread,10,main]

Answer: d

Explanation: Thread t has been made with default priority value 5 but in run method the priority has been explicitly changed to MAX_PRIORITY of class thread, that is 10 by code 't.setPriority(Thread.MAX_PRIORITY);' using the setPriority function of thread t.

Output:

```

$ javac multithreaded_programing.java
$ java multithreaded_programing
Thread[New Thread,10,main]

```

7. What will be the output of the following Java program?

```

1.     class newthread implements Runnable
2.     {
3.         Thread t;

```



```

4.         newthread()
5.     {
6.         t = new Thread(this, "My Thread");
7.         t.start();
8.     }
9. }
10. class multithreaded_programing
11. {
12.     public static void main(String args[])
13.     {
14.         new newthread();
15.     }
16. }

```

- a) My Thread
- b) Thread[My Thread,5,main]
- c) Compilation Error
- d) Runtime Error

Answer: c

Explanation: Thread t has been made by using Runnable interface, hence it is necessary to use inherited abstract method run() method to specify instructions to be implemented on the thread, since no run() method is used it gives a compilation error.

Output:

```

$ javac multithreaded_programing.java
The type newthread must implement the inherited abstract method Runnable.run()

```

8. What will be the output of the following Java program?

```

1.     class newthread implements Runnable
2.     {
3.         Thread t;
4.         newthread()
5.         {
6.             t = new Thread(this, "My Thread");
7.             t.start();
8.         }
9.         public void run()
10.        {
11.            System.out.println(t.getName());
12.        }

```

```

13.     }
14.     class multithreaded_programing
15.     {
16.         public static void main(String args[])
17.         {
18.             new newthread();
19.         }
20.     }

```

- a) My Thread
- b) Thread[My Thread,5,main]
- c) Compilation Error
- d) Runtime Error

Answer: a

Explanation: None.

Output:

```

$ javac multithreaded_programing.java
$ java multithreaded_programing
My Thread

```

9. What will be the output of the following Java program?

```

1.     class newthread implements Runnable
2.     {
3.         Thread t;
4.         newthread()
5.         {
6.             t = new Thread(this,"My Thread");
7.             t.start();
8.         }
9.         public void run()
10.        {
11.            System.out.println(t);
12.        }
13.    }
14.    class multithreaded_programing
15.    {
16.        public static void main(String args[])
17.        {

```

```
18.         new newthread();  
19.     }  
20. }
```

- a) My Thread
- b) Thread[My Thread,5,main]
- c) Compilation Error
- d) Runtime Error

Answer: b

Explanation: None.

Output:

```
$ javac multithreaded_programing.java  
$ java multithreaded_programing  
Thread[My Thread,5,main]
```

1. Which object Java application uses to create a new process?

- a) Process
- b) Builder
- c) ProcessBuilder
- d) CreateBuilder

Answer: c

Explanation: Java application uses ProcessBuilder object to create a new process. By default, same set of environment variables passed which are set in application's virtual machine process.

2. Which of the following is true about Java system properties?

- a) Java system properties are accessible by any process
- b) Java system properties are accessible by processes they are added to
- c) Java system properties are retrieved by System.getenv()
- d) Java system properties are set by System.setenv()

Answer: b

Explanation: Java system properties are only used and accessible by the processes they are added to.

Answer: a
Explanation: Java system properties can be set at runtime using System.setProperty(name, value) or using System.getProperties().load() methods.

4. Which system property stores installation directory of JRE?

- a) user.home
- b) java.class.path
- c) java.home
- d) user.dir

Answer: c

Explanation: java.home is the installation directory of Java Runtime Environment.

5. What does System.getProperty("variable") return?

- a) compilation error
- b) value stored in variable
- c) runtime error
- d) null

Answer: d

Explanation: System.getProperty("variable") returns null value. Because, variable is not a property and if property does not exist, this method returns null value.

6. What is true about the setProperties method?

- a) setProperties method changes the set of Java Properties which are persistent
- b) Changing the system properties within an application will affect future invocations
- c) setProperties method changes the set of Java Properties which are not persistent
- d) setProperties writes the values directly into the file which stores all the properties

Answer: c

Explanation: The changes made by the setProperties method are not persistent. Hence, it does not affect future invocation.

7. How to use environment properties in the class?

- a) @Environment
- b) @Variable
- c) @Property
- d) @Autowired

Answer: d

Explanation:

```
@Autowired
private Environment env;
```

This is how environment variables are injected in the class where they can be used.

9. Which environment variable is used to set java path?

- a) JAVA
- b) JAVA_HOME
- c) CLASSPATH
- d) MAVEN_HOME

Answer: b

Explanation: JAVA_HOME is used to store a path to the java installation.

10. How to read a classpath file?

- a) InputStream in = this.getClass().getResource("SomeTextFile.txt");
- b) InputStream in = this.getClass().getResourceClasspath("SomeTextFile.txt");
- c) InputStream in = this.getClass().getResourceAsStream("SomeTextFile.txt");
- d) InputStream in = this.getClass().getResource("classpath:/SomeTextFile.txt");

Answer: c

Explanation: This method can be used to load files using relative path to the package of the class.

1. Which of these is a process of writing the state of an object to a byte stream?

- a) Serialization
- b) Externalization
- c) File Filtering
- d) All of the mentioned

Answer: a

Explanation: Serialization is the process of writing the state of an object to a byte stream. This is used when you want to save the state of your program to a persistent storage area.

2. Which of these process occur automatically by the java runtime system?

- a) Serialization
- b) Garbage collection
- c) File Filtering
- d) All of the mentioned

Answer: a

Explanation: Serialization and deserialization occur automatically by java runtime system, Garbage collection also occur automatically but is done by CPU or the operating system not by the java runtime system.

3. Which of these is an interface for control over serialization and deserialization?

- a) Serializable
- b) Externalization
- c) FileFilter
- d) ObjectInput

Answer: b
Explanation: None.

4. Which of these interface extends DataOutput interface?

- a) Serializable
- b) Externalization
- c) ObjectOutput
- d) ObjectInput

Answer: c
Explanation: ObjectOutput interface extends the DataOutput interface and supports object serialization.

5. Which of these is a method of ObjectOutput interface used to finalize the output state so that any buffers are cleared?

- a) clear()
- b) flush()
- c) fflush()
- d) close()

Answer: b
Explanation: None.

6. Which of these is method of ObjectOutput interface used to write the object to input or output stream as required?

- a) write()
- b) Write()
- c) StreamWrite()
- d) writeObject()

Answer: d
Explanation: writeObject() is used to write an object into invoking stream, it can be input stream or output stream.

7. What will be the output of the following Java program?

```
1.      import java.io.*;
2.      class serialization
3.      {
4.          public static void main(String[] args)
5.          {
6.              try
7.              {
8.                  Myclass object1 = new Myclass("Hello", -7, 2.1e10);
9.                  FileOutputStream fos = new FileOutputStream("serial");
10.                 ObjectOutputStream oos = new ObjectOutputStream(fos);
```

```
11.         oos.writeObject(object1);
12.
13.         oos.flush();
14.
15.         oos.close();
16.     }
17.
18.     catch(Exception e)
19.     {
20.         System.out.println("Serialization" + e);
21.
22.         System.exit(0);
23.     }
24.
25.     try
26.     {
27.         Myclass object2;
28.
29.         FileInputStream fis = new FileInputStream("serial");
30.
31.         ObjectInputStream ois = new ObjectInputStream(fis);
32.
33.         object2 = (Myclass)ois.readObject();
34.
35.         ois.close();
36.
37.         System.out.println(object2);
38.     }
39.
40.     catch (Exception e)
41.     {
42.         System.out.print("deserialization" + e);
43.
44.         System.exit(0);
45.     }
46. }
47.
48. class Myclass implements Serializable
49. {
50.
51.     String s;
52.
53.     int i;
54.
55.     double d;
56.
57.     Myclass (String s, int i, double d)
58.     {
59.
60.         this.d = d;
61.
62.         this.i = i;
```

```
45.         this.s = s;
46.     }
47. }
```

- a) s=Hello; i=-7; d=2.1E10
- b) Hello; -7; 2.1E10
- c) s; i; 2.1E10
- d) Serialization

Answer: a

Explanation: None.

Output:

```
$ javac serialization.java
$ java serialization
s=Hello; i=-7; d=2.1E10
```

8. What will be the output of the following Java program?

```
1.     import java.io.*;
2.
3.     class serialization
4.     {
5.
6.         public static void main(String[] args)
7.         {
8.
9.             try
10.            {
11.
12.                Myclass object1 = new Myclass("Hello", -7, 2.1e10);
13.
14.                FileOutputStream fos = new FileOutputStream("serial");
15.
16.                ObjectOutputStream oos = new ObjectOutputStream(fos);
17.
18.                oos.writeObject(object1);
19.
20.                oos.flush();
21.
22.                oos.close();
23.
24.            }
25.
26.            catch(Exception e)
27.            {
28.
29.                System.out.println("Serialization" + e);
30.
31.                System.exit(0);
32.
33.            }
34.
35.        }
36.
37.        try
38.        {
39.
40.            int x;
```

```

23.         FileInputStream fis = new FileInputStream("serial");
24.         ObjectInputStream ois = new ObjectInputStream(fis);
25.         x = ois.readInt();
26.         ois.close();
27.         System.out.println(x);
28.     }
29.     catch (Exception e)
30.     {
31.         System.out.print("deserialization");
32.         System.exit(0);
33.     }
34. }
35. }
36. class Myclass implements Serializable
37. {
38.     String s;
39.     int i;
40.     double d;
41.     Myclass(String s, int i, double d)
42.     {
43.         this.d = d;
44.         this.i = i;
45.         this.s = s;
46.     }
47. }

```

- a) -7
- b) Hello
- c) 2.1E10
- d) deserialization

Answer: d

Explanation: $x = ois.readInt()$; will try to read an integer value from the stream 'serial' created before, since stream contains an object of Myclass hence error will occur and it will be caught by catch printing deserialization.

Output:

```

$ javac serialization.java
$ java serialization
deserialization

```

9. What will be the output of the following Java program?


```

1.     import java.io.*;
2.
3.     class Chararrayinput
4.     {
5.
6.         public static void main(String[] args)
7.         {
8.
9.             String obj  = "abcdefgh";
10.
11.             int length = obj.length();
12.
13.             char c[] = new char[length];
14.
15.             obj.getChars(0, length, c, 0);
16.
17.             CharArrayReader input1 = new CharArrayReader(c);
18.
19.             CharArrayReader input2 = new CharArrayReader(c, 1, 4);
20.
21.             int i;
22.
23.             int j;
24.
25.             try
26.             {
27.
28.                 while ((i = input1.read()) == (j = input2.read()))
29.                 {
30.
31.                     System.out.print((char)i);
32.
33.                 }
34.
35.             }
36.
37.             catch (IOException e)
38.             {
39.
40.                 e.printStackTrace();
41.
42.             }
43.
44.         }
45.
46.     }

```

- a) abc
- b) abcd
- c) abcde
- d) None of the mentioned

Answer: d

Explanation: No output is printed. CharArrayReader object input1 contains string “abcdefgh” whereas object input2 contains string “bcde”, when while((i=input1.read())==(j=input2.read())) is executed the starting character of each object is compared since they are unequal control comes out of loop and nothing is printed on the screen.

Output:

```

$ javac Chararrayinput.java
$ java Chararrayinput

```

10. What will be the output of the following Java program?

```
1.    import java.io.*;
2.
3.    class streams
4.    {
5.
6.        public static void main(String[] args)
7.        {
8.
9.            FileOutputStream fos = new FileOutputStream("serial");
10.           ObjectOutputStream oos = new ObjectOutputStream(fos);
11.           oos.writeFloat(3.5);
12.           oos.flush();
13.           oos.close();
14.       }
15.       catch(Exception e)
16.       {
17.           System.out.println("Serialization" + e);
18.           System.exit(0);
19.       }
20.       try
21.       {
22.           float x;
23.           FileInputStream fis = new FileInputStream("serial");
24.           ObjectInputStream ois = new ObjectInputStream(fis);
25.           x = ois.readInt();
26.           ois.close();
27.           System.out.println(x);
28.       }
29.       catch (Exception e)
30.       {
31.           System.out.print("deserialization");
32.           System.exit(0);
33.       }
34.   }
```

34. }

- a) 3
- b) 3.5
- c) serialization
- d) deserialization

Answer: b

Explanation: `oos.writeFloat(3.5);` writes in output stream which is extracted by `x = ois.readFloat();` and stored in `x` hence `x` contains 3.5.

Output:

```
$ javac streams.java
$ java streams
3.5
```

1. How an object can become serializable?

- a) If a class implements `java.io.Serializable` class
- b) If a class or any superclass implements `java.io.Serializable` interface
- c) Any object is serializable
- d) No object is serializable

Answer: b

Explanation: A Java object is serializable if class or any its superclass implements `java.io.Serializable` or its subinterface `java.io.Externalizable`.

2. What is serialization?

- a) Turning object in memory into stream of bytes
- b) Turning stream of bytes into an object in memory
- c) Turning object in memory into stream of bits
- d) Turning stream of bits into an object in memory

Answer: a

Explanation: Serialization in Java is the process of turning object in memory into stream of bytes.

3. What is deserialization?

- a) Turning object in memory into stream of bytes
- b) Turning stream of bytes into an object in memory
- c) Turning object in memory into stream of bits
- d) Turning stream of bits into an object in memory

Answer: b

Explanation: Deserialization is the reverse process of serialization which is turning stream of bytes into an object in memory.

4. How many methods `Serializable` has?

- a) 1
- b) 2
- c) 3
- d) 0

Answer: d

Explanation: `Serializable` interface does not have any method. It is also called a marker interface.

5. What type of members are not serialized?

- a) Private
- b) Protected
- c) Static

d) Throwable

Answer: c

Explanation: All static and transient variables are not serialized.

6. If member does not implement serialization, which exception would be thrown?

- a) RuntimeException**
- b) SerializableException**
- c) NotSerializableException**
- d) UnSerializedException**

Answer: c

Explanation: If member of a class does not implement serialization, NotSerializationException will be thrown. Answer: b

Explanation: Default serialization process can be overridden.

8. Which of the following methods is used to avoid serialization of new class whose super class already implements Serialization?

- a) writeObject()**
- b) readWriteObject()**
- c) writeReadObject()**
- d) unSerializaedObject()**

Answer: a

Explanation: writeObject() and readObject() methods should be implemented to avoid Java serialization.

9. Which of the following methods is not used while Serialization and DeSerialization?

- a) readObject()**
- b) readExternal()**
- c) readWriteObject()**
- d) writeObject()**

Answer: c

Explanation: Using readObject(), writeObject(), readExternal() and writeExternal() methods Serialization and DeSerialization are implemented. Answer: a

Explanation: Serialized object can be transferred via network because Java serialized object remains in form of bytes which can be transmitted over network.

1. Which of these is a process of extracting/removing the state of an object from a stream?

- a) Serialization**
- b) Externalization**
- c) File Filtering**
- d) Deserialization**

Answer: d

Explanation: Deserialization is a process by which the data written in the stream can be extracted out from the stream.

2. Which of these process occur automatically by java run time system?

- a) Serialization**
- b) Memory allocation**
- c) Deserialization**
- d) All of the mentioned**

Answer: d

Explanation: Serialization, deserialization and Memory allocation occur automatically by java run time system.

3. Which of these interface extends DataInput interface?

- a) Serializable**
- b) Externalization**

- c) ObjectOutputStream
- d) ObjectInput

Answer: d

Explanation: ObjectInput interface extends the DataInput interface and supports object serialization.

4. Which of these is a method of ObjectInput interface used to deserialize an object from a stream?

- a) int read()
- b) void close()
- c) Object readObject()
- d) Object WriteObject()

Answer: c

Explanation: None.

5. Which of these class extend InputStream class?

- a) ObjectStream
- b) ObjectInputStream
- c) ObjectOutput
- d) ObjectInput

Answer: b

Explanation: ObjectInputStream class extends the InputStream class and implements the ObjectInput interface.

6. What will be the output of the following Java code?

```
1.    import java.io.*;
2.
3.    class streams
4.    {
5.        public static void main(String[] args)
6.        {
7.            try
8.            {
9.                FileOutputStream fos = new FileOutputStream("serial");
10.               ObjectOutputStream oos = new ObjectOutputStream(fos);
11.               oos.writeInt(5);
12.               oos.flush();
13.               oos.close();
14.            }
15.            catch(Exception e)
16.            {
17.                System.out.println("Serialization" + e);
18.                System.exit(0);
19.            }
20.        }
21.    }
```

```

20.        {
21.            int z;
22.            FileInputStream fis = new FileInputStream("serial");
23.            ObjectInputStream ois = new ObjectInputStream(fis);
24.            z = ois.readInt();
25.            ois.close();
26.            System.out.println(x);
27.        }
28.        catch (Exception e)
29.        {
30.            System.out.print("deserialization");
31.            System.exit(0);
32.        }
33.    }
34. }

```

- a) 5
- b) void
- c) serialization
- d) deserialization

Answer: a

Explanation: oos.writeInt(5); writes integer 5 in the Output stream which is extracted by z = ois.readInt(); and stored in z hence z contains 5.

Output:

```

$ javac streams.java
$ java streams
5

```

7. What will be the output of the following Java code?

```

1.    import java.io.*;
2.
3.    class serialization
4.    {
5.
6.        public static void main(String[] args)
7.        {
8.
9.            Myclass object1 = new Myclass("Hello", -7, 2.1e10);
10.           FileOutputStream fos = new FileOutputStream("serial");
11.           ObjectOutputStream oos = new ObjectOutputStream(fos);

```

```
11.         oos.writeObject(object1);
12.
13.         oos.flush();
14.
15.         oos.close();
16.     }
17.
18.     catch(Exception e)
19.     {
20.         System.out.println("Serialization" + e);
21.
22.         System.exit(0);
23.     }
24.
25.     try
26.     {
27.         int x;
28.
29.         FileInputStream fis = new FileInputStream("serial");
30.
31.         ObjectInputStream ois = new ObjectInputStream(fis);
32.
33.         x = ois.readInt();
34.
35.         ois.close();
36.
37.         System.out.println(x);
38.     }
39.
40.     catch (Exception e)
41.     {
42.         System.out.print("deserialization");
43.
44.         System.exit(0);
45.     }
46. }
47.
48. class Myclass implements Serializable
49. {
50.
51.     String s;
52.
53.     int i;
54.
55.     double d;
56.
57.     Myclass(String s, int i, double d)
58.     {
59.
60.         this.d = d;
61.
62.         this.i = i;
```

```
45.         this.s = s;
46.     }
47. }
```

- a) -7
- b) Hello
- c) 2.1E10
- d) deserialization

Answer: d

Explanation: $x = ois.readInt()$; will try to read an integer value from the stream 'serial' created before, since stream contains an object of MyClass hence error will occur and it will be caught by catch printing deserialization.

Output:

```
$ javac serialization.java
$ java serialization
deserialization
```

8. What will be the output of the following Java program?

```
1.     import java.io.*;
2.
3.     class streams
4.     {
5.
6.         public static void main(String[] args)
7.         {
8.
9.             try
10.            {
11.                FileOutputStream fos = new FileOutputStream("serial");
12.                ObjectOutputStream oos = new ObjectOutputStream(fos);
13.                oos.writeFloat(3.5);
14.                oos.flush();
15.                oos.close();
16.            }
17.            catch(Exception e)
18.            {
19.                System.out.println("Serialization" + e);
20.                System.exit(0);
21.            }
22.        }
23.
24.        try
25.        {
26.            FileInputStream fis = new FileInputStream("serial");
27.            ObjectInputStream ois = new ObjectInputStream(fis);
```



```

23.         ois.close();
24.         System.out.println(ois.available());
25.     }
26.     catch (Exception e)
27.     {
28.         System.out.print("deserialization");
29.         System.exit(0);
30.     }
31. }
32. }

```

- a) 1
- b) 2
- c) 3
- d) 0

Answer: d

Explanation: New input stream is linked to steal 'serials', an object 'ois' of ObjectOutputStream is used to access this newly created stream, ois.close(); closes the stream hence we can't access the stream and ois.available() returns 0.

Output:

```

$ javac streams.java
$ java streams
0

```

9. What will be the output of the following Java program?

```

1.     import java.io.*;
2.     class streams
3.     {
4.         public static void main(String[] args)
5.         {
6.             try
7.             {
8.                 FileOutputStream fos = new FileOutputStream("serial");
9.                 ObjectOutputStream oos = new ObjectOutputStream(fos);
10.                oos.writeFloat(3.5);
11.                oos.flush();
12.                oos.close();
13.            }
14.            catch(Exception e)
15.            {

```

```

16.         System.out.println("Serialization" + e);
17.
18.         System.exit(0);
19.     }
20.
21.     try
22.     {
23.         FileInputStream fis = new FileInputStream("serial");
24.         ObjectInputStream ois = new ObjectInputStream(fis);
25.         System.out.println(ois.available());
26.     }
27.     catch (Exception e)
28.     {
29.         System.out.print("deserialization");
30.         System.exit(0);
31.     }

```

- a) 1
- b) 2
- c) 3
- d) 4

Answer: d

Explanation: oos.writeFloat(3.5); writes 3.5 in output stream. A new input stream is linked to stream 'serials', an object 'ois' of ObjectInputStream is used to access this newly created stream, ois.available() gives the total number of byte in the input stream since a float was written in the stream thus the stream contains 4 byte, hence 4 is returned and printed.

Output:

```

$ javac streams.java
$ java streams
4

```

10. What will be the output of the following Java program?

```

1. import java.io.FileOutputStream;
2. public class FileOutputStreamExample
3. {
4.     public static void main(String args[])
5.     {
6.         try
7.         {
8.             FileOutputStream fout=new FileOutputStream("D:\\sanfoundry.txt");

```

```

9.         String s="Welcome to Sanfoundry.";
10.        byte b[]=s.getBytes();//converting string into byte array
11.        fout.write(b);
12.        fout.close();
13.        System.out.println("Success");
14.    }    catch(Exception e){System.out.println(e);}
15.    }
16. }

```

- a) “Success” to the output and “Welcome to Sanfoundry” to the file
- b) only “Welcome to Sanfoundry” to the file
- c) compile time error
- d) No Output

Answer: a

Explanation: First, it will print “Success” and besides that it will write “Welcome to Sanfoundry” to the file sanfoundry.txt.

1. Which of these package contains classes and interfaces for networking?

- a) java.io
- b) java.util
- c) java.net
- d) java.network

Answer: c

Explanation: None.

2. Which of these is a protocol for breaking and sending packets to an address across a network?

- a) TCP/IP
- b) DNS
- c) Socket
- d) Proxy Server

Answer: a

Explanation: TCP/IP – Transfer control protocol/Internet Protocol is used to break data into small packets and send them to an address across a network.

3. How many ports of TCP/IP are reserved for specific protocols?

- a) 10
- b) 1024
- c) 2048
- d) 512

Answer: b

Explanation: None.

4. How many bits are in a single IP address?

- a) 8
- b) 16
- c) 32
- d) 64

Answer: c

Explanation: None.

5. Which of these is a full form of DNS?

- a) Data Network Service
- b) Data Name Service
- c) Domain Network Service
- d) Domain Name Service

Answer: d

Explanation: None.

6. Which of these class is used to encapsulate IP address and DNS?

- a) DatagramPacket
- b) URL
- c) InetAddress
- d) ContentHandler

Answer: c

Explanation: InetAddress class encapsulate both IP address and DNS, we can interact with this class by using name of an IP host.

7. What will be the output of the following Java program?

```
1.      import java.net.*;
2.
3.      class networking
4.      {
5.
6.          public static void main(String[] args) throws UnknownHostException
7.          {
8.
9.              InetAddress obj1 = InetAddress.getByName("sanfoundry.com");
10.             InetAddress obj2 = InetAddress.getByName("sanfoundry.com");
11.             boolean x = obj1.equals(obj2);
12.             System.out.print(x);
13.         }
14.     }
```

- a) 0
- b) 1
- c) true
- d) false

Answer: c

Explanation: None.

Output:

```
$ javac networking.java
$ java networking
true
```

8. What will be the output of the following Java program?

```
1.      import java.net.*;
```

```

2.     public class networking
3.     {
4.         public static void main(String[] args) throws UnknownHostException
5.         {
6.             InetAddress obj1 = InetAddress.getByName("cisco.com");
7.             InetAddress obj2 = InetAddress.getByName("sanfoundry.com");
8.             boolean x = obj1.equals(obj2);
9.             System.out.print(x);
10.        }
11.    }

```

- a) 0
- b) 1
- c) true
- d) false

Answer: d

Explanation: `InetAddress obj1 = InetAddress.getByName("cisco.com");` creates object `obj1` having DNS and IP address of `cisco.com`, `InetAddress obj2 = InetAddress.getByName("sanfoundry.com");` creates `obj2` having DNS and IP address of `sanfoundry.com`, since both these address point to two different locations `false` is returned by `obj1.equals(obj2);`.

Output:

```

$ javac networking.java
$ java networking
false

```

9. What will be the output of the following Java program?

```

1. import java.io.*;
2. import java.net.*;
3. public class URLEDemo
4. {
5.     public static void main(String[] args)
6.     {
7.         try
8.         {
9.             URL url=new URL("https://www.sanfoundry.com/java-mcq");
10.            System.out.println("Protocol: "+url.getProtocol());
11.            System.out.println("Host Name: "+url.getHost());
12.            System.out.println("Port Number: "+url.getPort());
13.        } catch(Exception e){System.out.println(e);}
14.    }

```

15. }

- a) Protocol: http
- b) Host Name: www.sanfoundry.com
- c) Port Number: -1
- d) All of the mentioned

Answer: d

Explanation: `getProtocol()` give protocol which is http

`getUrl()` give name domain name

`getPort()` Since we have not explicitly set the port, default value that is -1 is printed.

10. What will be the output of the following Java program?

```
1.      import java.net.*;
2.
3.      class networking
4.      {
5.
6.          public static void main(String[] args) throws UnknownHostException
7.          {
8.
9.              InetAddress obj1 = InetAddress.getByName("cisco.com");
10.             System.out.print(obj1.getHostName());
11.
12.          }
13.      }
```

- a) cisco
- b) cisco.com
- c) www.cisco.com
- d) none of the mentioned

Answer: b

Explanation: None.

Output:

```
$ javac networking.java
$ java networking
cisco.com
```

1. Which of these interface abstractes the output of messages from httpd?

- a) LogMessage
- b) LogResponse
- c) Httpdserver
- d) httpdResponse

Answer: a

Explanation: LogMessage is a simple interface that is used to abstract the output of messages from the httpd.

2. Which of these class is used to create servers that listen for either local or remote client programs?

- a) httpServer
- b) ServerSockets
- c) MimeHeader
- d) HttpResponse

Answer: b

Explanation: None.

3. Which of these is a standard for communicating multimedia content over email?

- a) http
- b) https
- c) Mime
- d) httpd

Answer: c

Explanation: MIME is an internet standard for communicating multimedia content over email. The HTTP protocol uses and extends the notion of MIME headers to pass attribute pairs between HTTP client and server.

4. Which of these methods is used to make raw MIME formatted string?

- a) parse()
- b) toString()
- c) getString()
- d) parseString()

Answer: a

Explanation: None.

5. Which of these class is used for operating on request from the client to the server?

- a) http
- b) httpDecoder
- c) httpConnection
- d) httpd

Answer: d

Explanation: None.

6. Which of these method of MimeHeader is used to return the string equivalent of the values stores on MimeHeader?

- a) string()
- b) toString()
- c) convertString()
- d) getString()

Answer: b

Explanation: toString() does the reverse of parse() method, it is used to return the string equivalent of the values stores on MimeHeader.

7. What will be the output of the following Java program?

```
1.    import java.net.*;
2.
3.    class networking
4.    {
5.
6.        public static void main(String[] args) throws Exception
7.        {
8.
9.            URL obj = new URL("https://www.sanfoundry.com/javamcq");
10.
11.            URLConnection obj1 = obj.openConnection();
12.
13.            System.out.print(obj1.getContentType());
14.
15.        }
```

10. }

Note: Host URL is written in html and simple text.

- a) html
- b) text
- c) html/text
- d) text/html

Answer: d

Explanation: None.

Output:

```
$ javac networking.java
$ java networking
text/html
```

8. Which of these is an instance variable of class httpd?

- a) port
- b) cache
- c) log
- d) All of the mentioned

Answer: d

Explanation: There are 5 instance variables: port, docRoot, log, cache and stopFlag. All of them are private.

9. What will be the output of the following Java program?

```
1.    import java.net.*;
2.
3.    class networking
4.    {
5.
6.        public static void main(String[] args) throws MalformedURLException
7.        {
8.
9.            URL obj = new URL("https://www.sanfoundry.com/javamcq");
10.           System.out.print(obj.toExternalForm());
11.
12.        }
13.    }
```

- a) sanfoundry
- b) sanfoundry.com
- c) www.sanfoundry.com
- d) https://www.sanfoundry.com/javamcq

Answer: d

Explanation: toExternalForm() is used to know the full URL of an URL object.

Output:

```
$ javac networking.java
$ java networking
https://www.sanfoundry.com/javamcq
```

1. Which of these methods of httpd class is used to read data from the stream?

- a) getDta()
- b) GetResponse()
- c) getStream()

d) getRawRequest()

Answer: d

Explanation: The getRawRequest() method reads data from a stream until it gets two consecutive newline characters.

2. Which of these method of httpd class is used to get report on each hit to HTTP server?

- a) log()
- b) logEntry()
- c) logHttpd()
- d) logResponse()

Answer: b

Explanation: None.

3. Which of these methods are used to find a URL from the cache of httpd?

- a) findfromCache()
- b) findFromCache()
- c) serveFromCache()
- d) getFromCache()

Answer: c

Explanation: serveFromCache() is a boolean method that attempts to find a particular URL in the cache. If it is successful then the content of that cache entry are written to the client, otherwise it returns false.

4. Which of these variables stores the number of hits that are successfully served out of cache?

- a) hits
- b) hitstocache
- c) hits_to_cache
- d) hits.to.cache

Answer: d

Explanation: None.

5. Which of these method of httpd class is used to write UrlCacheEntry object into local disk?

- a) writeDiskCache()
- b) writetoDisk()
- c) writeCache()
- d) writeDiskEntry()

Answer: a

Explanation: The writeDiskCache() method takes an UrlCacheEntry object and writes it persistently into the local disk. It constructs directory names out of URL, making sure to replace the slash(/) characters with system dependent separatorChar.

6. What will be the output of the following Java program?

```
1.      import java.net.*;
2.
3.      class networking
4.      {
5.
6.          public static void main(String[] args) throws Exception
7.          {
8.
9.              URL obj = new URL("https://www.sanfoundry.com/javamcq");
10.
11.              URLConnection obj1 = obj.openConnection();
```

```
8.         int len = obj1.getContentLength();  
9.         System.out.print(len);  
10.    }  
11. }
```

Note: Host URL is having length of content 127.

- a) 126
- b) 127
- c) Compilation Error
- d) Runtime Error

Answer: b

Explanation: None.

Output:

```
$ javac networking.java  
$ java networking  
127
```

7. Which of these method is used to start a server thread?

- a) run()
- b) start()
- c) runThread()
- d) startThread()

Answer: a

Explanation: run() method is called when the server thread is started.

8. Which of these method is called when http daemon is acting like a normal web server?

- a) Handle()
- b) HandleGet()
- c) handleGet()
- d) Handleget()

Answer: c

Explanation: None.

1. What does URL stands for?

- a) Uniform Resource Locator
- b) Uniform Resource Latch
- c) Universal Resource Locator
- d) Universal Resource Latch

Answer: a

Explanation: URL is Uniform Resource Locator.

2. Which of these exceptions is thrown by URL class's constructors?

- a) URLNotFound
- b) URLSourceNotFound
- c) MalformedURLException
- d) URLNotFoundExpection

Answer: c

Explanation: None.

3. Which of these methods is used to know host of an URL?

- a) host()

- b) `getHost()`
- c) `GetHost()`
- d) `gethost()`

Answer: b

Explanation: None.

4. Which of these methods is used to know the full URL of an URL object?

- a) `fullHost()`
- b) `getHost()`
- c) `ExternalForm()`
- d) `toExternalForm()`

Answer: d

Explanation: None.

5. Which of these class is used to access actual bits or content information of a URL?

- a) `URL`
- b) `URLDecoder`
- c) `URLConnection`
- d) All of the mentioned

Answer: d

Explanation: URL, URLDecoder and URLConnection all there are used to access information stored in a URL.

6. What will be the output of the following Java code?

```
1.      import java.net.*;
2.
3.      class networking
4.      {
5.
6.          public static void main(String[] args) throws MalformedURLException
7.          {
8.              URL obj = new URL("https://www.sanfoundry.com/javamcq");
9.              System.out.print(obj.getProtocol());
10.         }
11.     }
```

- a) `http`
- b) `https`
- c) `www`
- d) `com`

Answer: a

Explanation: obj.getProtocol() is used to know the protocol used by the host. http stands for hypertext transfer protocol, usually 2 types of protocols are used http and https, where s in https stands for secured.

Output:

```
$ javac networking.java
$ java networking
http
```

7. What will be the output of the following Java program?

```

1.     import java.net.*;
2.
3.     class networking
4.     {
5.
6.         public static void main(String[] args) throws MalformedURLException
7.         {
8.             URL obj = new URL("https://www.sanfoundry.com/javamcq");
9.             System.out.print(obj.getPort());
10.        }
11.    }

```

- a) 1
- b) 0
- c) -1
- d) garbage value

Answer: c

Explanation: Since we have not explicitly set the port default value that is -1 is printed.

Output:

```

$ javac networking.java
$ java networking
-1

```

8. What will be the output of the following Java program?

```

1.     import java.net.*;
2.
3.     class networking
4.     {
5.
6.         public static void main(String[] args) throws MalformedURLException
7.         {
8.             URL obj = new URL("https://www.sanfoundry.com/javamcq");
9.             System.out.print(obj.getHost());
10.        }
11.    }

```

- a) sanfoundry
- b) sanfoundry.com
- c) www.sanfoundry.com
- d) https://www.sanfoundry.com/javamcq

Answer: c

Explanation: None.

Output:

```

$ javac networking.java
$ java networking
www.sanfoundry.com

```

9. What will be the output of the following Java program?

```
1.    import java.net.*;
2.
3.    class networking
4.    {
5.
6.        public static void main(String[] args) throws MalformedURLException
7.        {
8.
9.            URL obj = new URL("https://www.sanfoundry.com/javamcq");
10.
11.            System.out.print(obj.toExternalForm());
12.
13.        }
14.    }
```

- a) sanfoundry
- b) sanfoundry.com
- c) www.sanfoundry.com
- d) https://www.sanfoundry.com/javamcq

Answer: d

Explanation: toExternalForm() is used to know the full URL of an URL object.

Output:

```
$ javac networking.java
$ java networking
https://www.sanfoundry.com/javamcq
```

1. Which of these is a wrapper around everything associated with a reply from an http server?

- a) HTTP
- b) HttpResponse
- c) HttpRequest
- d) httpserver

Answer: b

Explanation: HttpResponse is wrapper around everything associated with a reply from an http server.

2. Which of these transfer protocol must be used so that URL can be accessed by URLConnection class object?

- a) http
- b) https
- c) Any Protocol can be used
- d) None of the mentioned

Answer: a

Explanation: For a URL to be accessed from remote location http protocol must be used.

3. Which of these methods is used to know when was the URL last modified?

- a) LastModified()
- b) getLastModified()
- c) GetLastModified()
- d) getlastModified()

Answer: b

Explanation: None.

4. Which of these methods is used to know the type of content used in the URL?

- a) `ContentType()`
- b) `contentType()`
- c) `getContentType()`
- d) `GetContentType()`

Answer: c

Explanation: None.

5. Which of these data member of `HttpResponse` class is used to store the response from an http server?

- a) `status`
- b) `address`
- c) `statusResponse`
- d) `statusCode`

Answer: d

Explanation: When we send a request to an http server it responds with a status code this status code is stored in `statusCode` and a textual equivalent which is stored in `reasonPhrase`.

6. What will be the output of the following Java program?

```
1.      import java.net.*;
2.
3.      class networking
4.      {
5.
6.          public static void main(String[] args) throws Exception
7.          {
8.
9.              URL obj = new URL("https://www.sanfoundry.com/javamcq");
10.             URLConnection obj1 = obj.openConnection();
11.
12.             System.out.print(obj1.getContentType());
13.
14.         }
15.     }
```

Note: Host URL is written in html and simple text.

- a) `html`
- b) `text`
- c) `html/text`
- d) `text/html`

Answer: d

Explanation: None.

Output:

```
$ javac networking.java
$ java networking
text/html
```

7. What will be the output of the following Java program?

```
1.      import java.net.*;
2.
3.      class networking
4.      {
```

```

4.         public static void main(String[] args) throws Exception
5.     {
6.         URL obj = new URL("https://www.sanfoundry.com/javamcq");
7.         URLConnection obj1 = obj.openConnection();
8.         int len = obj1.getContentLength();
9.         System.out.print(len);
10.    }
11. }

```

Note: Host URL is having length of content 127.

a) 126

b) 127

c) Compilation Error

d) Runtime Error

Answer: b

Explanation: None.

Output:

```

$ javac networking.java
$ java networking
127

```

8. What will be the output of the following Java program?

```

1.     import java.net.*;
2.     class networking
3.     {
4.         public static void main(String[] args) throws Exception
5.     {
6.         URL obj = new URL("https://www.sanfoundry.com/javamcq");
7.         URLConnection obj1 = obj.openConnection();
8.         System.out.print(obj1.getLastModified);
9.     }
10. }

```

Note: Host URL was last modified on july 18 tuesday 2013 .

a) july

b) 18-6-2013

c) Tue 18 Jun 2013

d) Tue Jun 18 2013

Answer: d

Explanation: None.

Output:

```

$ javac networking.java
$ java networking

```

1. Which of these is a bundle of information passed between machines?

- a) Mime**
- b) Cache**
- c) Datagrams**
- d) DatagramSocket**

Answer: c

Explanation: The Datagrams are the bundle of information passed between machines.

2. Which of these class is necessary to implement datagrams?

- a) DatagramPacket**
- b) DatagramSocket**
- c) All of the mentioned**
- d) None of the mentioned**

Answer: c

Explanation: None.

3. Which of these method of DatagramPacket is used to find the port number?

- a) port()**
- b) getPort()**
- c) findPort()**
- d) recievePort()**

Answer: b

Explanation: None.

4. Which of these method of DatagramPacket is used to obtain the byte array of data contained in a datagram?

- a) getData()**
- b) getBytes()**
- c) getArray()**
- d) recieveBytes()**

Answer: a

Explanation: None.

5. Which of these methods of DatagramPacket is used to find the length of byte array?

- a) getnumber()**
- b) length()**
- c) Length()**
- d) getLength()**

Answer: d

Explanation: getLength returns the length of the valid data contained in the byte array that would be returned from the getData () method. This typically is not equal to length of whole byte array.

6. Which of these class must be used to send a datagram packets over a connection?

- a) InetAdress**
- b) DatagramPacket**
- c) DatagramSocket**
- d) All of the mentioned**

Answer: d

Explanation: By using 5 classes we can send and receive data between client and server; these are InetAddress, Socket, ServerSocket, DatagramSocket, and DatagramPacket.

7. Which of these method of DatagramPacket class is used to find the destination address?

- a) findAddress()
- b) getAddress()
- c) Address()
- d) whois()

Answer: b

Explanation: None.

8. Which of these is a return type of getAddress() method of DatagramPacket class?

- a) DatagramPacket
- b) DatagramSocket
- c) InetAddress
- d) ServerSocket

Answer: c

Explanation: None.

9. Which API gets the SocketAddress (usually IP address + port number) of the remote host that this packet is being sent to or is coming from.

- a) getSocketAddress()
- b) getAddress()
- c) address()
- d) none of the mentioned

Answer: a

Explanation: getSocketAddress() is used to get the socket address.

1. Which of these standard collection classes implements a dynamic array?

- a) AbstractList
- b) LinkedList
- c) ArrayList
- d) AbstractSet

Answer: c

Explanation: ArrayList class implements a dynamic array by extending AbstractList class.

2. Which of these class can generate an array which can increase and decrease in size automatically?

- a) ArrayList()
- b) DynamicList()
- c) LinkedList()
- d) MallocList()

Answer: a

Explanation: None.

3. Which of these method can be used to increase the capacity of ArrayList object manually?

- a) Capacity()
- b) increaseCapacity()
- c) increasecapacity()
- d) ensureCapacity()

Answer: d

Explanation: When we add an element, the capacity of ArrayList object increases automatically, but we can increase it manually to specified length x by using function ensureCapacity(x);

4. Which of these method of ArrayList class is used to obtain present size of an object?

- a) size()

- b) `length()`
- c) `index()`
- d) `capacity()`

Answer: a

Explanation: None.

5. Which of these methods can be used to obtain a static array from an `ArrayList` object?

- a) `Array()`
- b) `covertArray()`
- c) `toArray()`
- d) `coverttoArray()`

Answer: c

Explanation: None.

6. Which of these method is used to reduce the capacity of an `ArrayList` object?

- a) `trim()`
- b) `trimSize()`
- c) `trimTosize()`
- d) `trimToSize()`

Answer: d

Explanation: `trimTosize()` is used to reduce the size of the array that underlines an `ArrayList` object.

7. What will be the output of the following Java program?

```
1.      import java.util.*;
2.
3.      class Arraylist
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              ArrayList obj = new ArrayList();
10.
11.              obj.add("A");
12.
13.              obj.add("B");
14.
15.              obj.add("C");
16.
17.              obj.add(1, "D");
18.
19.              System.out.println(obj);
20.
21.          }
22.      }
```

- a) [A, B, C, D]
- b) [A, D, B, C]
- c) [A, D, C]
- d) [A, B, C]

Answer: b

Explanation: `obj` is an object of class `ArrayList` hence it is an dynamic array which can increase and decrease its size. `obj.add("X")` adds to the array element `X` and `obj.add(1,"X")` adds element `x` at index position `1` in the list, Hence `obj.add(1,"D")` stores `D` at index position `1` of `obj` and shifts the previous value stored at that position by `1`.

Output:

```
$ javac ArrayList.java
$ java ArrayList
[A, D, B, C].
```

8. What will be the output of the following Java program?

```
1.      import java.util.*;
2.
3.      class Output
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              ArrayList obj = new ArrayList();
10.             obj.add("A");
11.             obj.add(0, "B");
12.             System.out.println(obj.size());
13.         }
14.     }
```

- a) 0
- b) 1
- c) 2
- d) Any Garbage Value

Answer: c

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
2
```

9. What will be the output of the following Java program?

```
1.      import java.util.*;
2.
3.      class Output
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              ArrayList obj = new ArrayList();
10.             obj.add("A");
11.             obj.ensureCapacity(3);
12.             System.out.println(obj.size());
13.         }
14.     }
```

```
11.    }
```

- a) 1
- b) 2
- c) 3
- d) 4

Answer: a

Explanation: Although `obj.ensureCapacity(3);` has manually increased the capacity of `obj` to 3 but the value is stored only at index 0, therefore `obj.size()` returns the total number of elements stored in the `obj` i.e 1, it has nothing to do with `ensureCapacity()`.

Output:

```
$ javac Output.java
$ java Output
1
```

10. What will be the output of the following Java program?

```
1.    class Output
2.    {
3.        public static void main(String args[])
4.        {
5.            ArrayList obj = new ArrayList();
6.            obj.add("A");
7.            obj.add("D");
8.            obj.ensureCapacity(3);
9.            obj.trimToSize();
10.           System.out.println(obj.size());
11.        }
12.    }
```

- a) 1
- b) 2
- c) 3
- d) 4

Answer: b

Explanation: `trimToSize()` is used to reduce the size of the array that underlines an `ArrayList` object.

Output:

```
$ javac Output.java
$ java Output
2
```

Answer: b

Explanation: `Collection` interface provides `add`, `remove`, `search` or `iterate` while `map` has `clear`, `get`, `put`, `remove`, etc.

2. Which of the below does not implement Map interface?

- a) `HashMap`
- b) `Hashtable`
- c) `EnumMap`

d) Vector

Answer: d

Explanation: Vector implements AbstractList which internally implements Collection. Others come from implementing the Map interface.

3. What is the premise of equality for IdentityHashMap?

- a) Reference equality**
- b) Name equality**
- c) Hashcode equality**
- d) Length equality**

Answer: a

Explanation: IdentityHashMap is rarely used as it violates the basic contract of implementing equals() and hashCode() method.

4. What happens if we put a key object in a HashMap which exists?

- a) The new object replaces the older object**
- b) The new object is discarded**
- c) The old object is removed from the map**
- d) It throws an exception as the key already exists in the map**

Answer: a

Explanation: HashMap always contains unique keys. If same key is inserted again, the new object replaces the previous object.

5. While finding the correct location for saving key value pair, how many times the key is hashed?

- a) 1**
- b) 2**
- c) 3**
- d) unlimited till bucket is found**

Answer: b

Explanation: The key is hashed twice; first by hashCode() of Object class and then by internal hashing method of HashMap class.

Answer: b
Explanation: Hashmap outputs in the order of hashcode of the keys. So it is unordered but will always have same result for same set of keys.

7. If two threads access the same hashmap at the same time, what would happen?

- a) ConcurrentModificationException**
- b) NullPointerException**
- c) ClassNotFoundException**
- d) RuntimeException**

Answer: a

Explanation: The code will throw ConcurrentModificationException if two threads access the same hashmap at the same time.

Answer: c
Explanation: Collections.synchronizedMap() synchronizes entire map. ConcurrentHashMap provides thread safety without synchronizing entire map.

9. What will be the output of the following Java code snippet?

```
1. public class Demo
2. {
3.     public static void main(String[] args)
4.     {
```

```

5.         Map<Integer, Object> sampleMap = new TreeMap<Integer, Object>();
6.
7.         sampleMap.put(1, null);
8.
9.         sampleMap.put(5, null);
10.
11.        sampleMap.put(3, null);
12.
13.        sampleMap.put(2, null);
14.
15.        sampleMap.put(4, null);
16.
17.        System.out.println(sampleMap);
18.    }
19. }

```

- a) {1=null, 2=null, 3=null, 4=null, 5=null}
- b) {5=null}
- c) Exception is thrown
- d) {1=null, 5=null, 3=null, 2=null, 4=null}

Answer: a

Explanation: HashMap needs unique keys. TreeMap sorts the keys while storing objects.

10. If large number of items are stored in hash bucket, what happens to the internal structure?

- a) The bucket will switch from LinkedList to BalancedTree
- b) The bucket will increase its size by a factor of load size defined
- c) The LinkedList will be replaced by another hashmap
- d) Any further addition throws Overflow exception

Answer: a

Explanation: BalancedTree will improve performance from $O(n)$ to $O(\log n)$ by reducing hash collisions.

1. How can we remove an object from ArrayList?

- a) remove() method
- b) using Iterator
- c) remove() method and using Iterator
- d) delete() method

Answer: c

Explanation: There are 2 ways to remove an object from ArrayList. We can use overloaded method remove(int index) or remove(Object obj). We can also use an Iterator to remove the object.

2. How to remove duplicates from List?

- a) HashSet<String> listToSet = new HashSet<String>(duplicateList);
- b) HashSet<String> listToSet = duplicateList.toSet();
- c) HashSet<String> listToSet = Collections.convertToSet(duplicateList);
- d) HashSet<String> listToSet = duplicateList.getSet();

Answer: a

Explanation: Duplicate elements are allowed in List. Set contains unique objects.

3. How to sort elements of ArrayList?

- a) Collection.sort(listObj);
- b) Collections.sort(listObj);
- c) listObj.sort();
- d) SortersortAsc(listObj);

Answer: b
Explanation: Collections provides a method to sort the list. The order of sorting can be defined using Comparator.

- 4. When two threads access the same ArrayList object what is the outcome of the program?**
- a) Both are able to access the object**
 - b) ConcurrentModificationException is thrown**
 - c) One thread is able to access the object and second thread gets Null Pointer exception**
 - d) One thread is able to access the object and second thread will wait till control is passed to the second one**

Answer: b
Explanation: ArrayList is not synchronized. Vector is the synchronized data structure.

- 5. How is Arrays.asList() different than the standard way of initialising List?**
- a) Both are same**
 - b) Arrays.asList() throws compilation error**
 - c) Arrays.asList() returns a fixed length list and doesn't allow to add or remove elements**
 - d) We cannot access the list returned using Arrays.asList()**

Answer: c
Explanation: List returned by Arrays.asList() is a fixed length list which doesn't allow us to add or remove element from it.add() and remove() method will throw UnsupportedOperationException if used.

- 6. What is the difference between length() and size() of ArrayList?**
- a) length() and size() return the same value**
 - b) length() is not defined in ArrayList**
 - c) size() is not defined in ArrayList**
 - d) length() returns the capacity of ArrayList and size() returns the actual number of elements stored in the list**

Answer: d
Explanation: length() returns the capacity of ArrayList and size() returns the actual number of elements stored in the list which is always less than or equal to capacity.

- 7. Which class provides thread safe implementation of List?**
- a) ArrayList**
 - b) CopyOnWriteArrayList**
 - c) HashList**
 - d) List**

Answer: b
Explanation: CopyOnWriteArrayList is a concurrent collection class. Its very efficient if ArrayList is mostly used for reading purpose because it allows multiple threads to read data without locking, which was not possible with synchronized ArrayList.

- 8. Which of the below is not an implementation of List interface?**
- a) RoleUnresolvedList**
 - b) Stack**
 - c) AtttributeList**
 - d) SessionList**

Answer: d
Explanation: SessionList is not an implementation of List interface. The others are concrete classes of List.

- 9. What is the worst case complexity of accessing an element in ArrayList?**
- a) O(n)**
 - b) O(1)**
 - c) O(nlogn)**
 - d) O(2)**

Answer: b

Explanation: ArrayList has $O(1)$ complexity for accessing an element in ArrayList. $O(n)$ is the complexity for accessing an element from LinkedList. Answer: a

Explanation: If we make a copy of array before any changes to the array the content will not change. Else the content of the array will undergo changes.

```
1. public void setMyArray(String[] myArray)
2. {
3.     if(myArray == null)
4.     {
5.         this.myArray = new String[0];
6.     }
7.     else
8.     {
9.         this.myArray = Arrays.copyOf(newArray, newArray.length);
10.    }
11. }
```

1. What is the default clone of HashSet?

- a) Deep clone
- b) Shallow clone
- c) Plain clone
- d) Hollow clone

Answer: b

Explanation: Default clone() method uses shallow copy. The internal elements are not cloned. A shallow copy only copies the reference object. Answer: b

Explanation: get(Object o) method is useful when we want to compare objects based on the comparison of values.

HashSet does not provide any way to compare objects. It just guarantees unique objects stored in the collection.

3. What does Collections.emptySet() return?

- a) Immutable Set
- b) Mutable Set
- c) The type of Set depends on the parameter passed to the emptySet() method
- d) Null object

Answer: a

Explanation: Immutable Set is useful in multithreaded environment. One does not need to declare generic type collection. It is inferred by the context of method call.

4. What are the initial capacity and load factor of HashSet?

- a) 10, 1.0
- b) 32, 0.75
- c) 16, 0.75
- d) 32, 1.0

Answer: c

Explanation: We should not set the initial capacity too high and load factor too low if iteration performance is needed.

5. What is the relation between hashset and hashmap?

- a) HashSet internally implements HashMap

- b) HashMap internally implements HashSet
- c) HashMap is the interface; HashSet is the concrete class
- d) HashSet is the interface; HashMap is the concrete class

Answer: a

Explanation: HashSet is implemented to provide uniqueness feature which is not provided by HashMap. This also reduces code duplication and provides the memory efficient behavior of HashMap.

6. What will be the output of the following Java code snippet?

```
1. public class Test
2. {
3.     public static void main(String[] args)
4.     {
5.         Set s = new HashSet();
6.         s.add(new Long(10));
7.         s.add(new Integer(10));
8.         for(Object object : s)
9.         {
10.             System.out.println("test - "+object);
11.         }
12.     }
13. }
```

Answer: a

Explanation: Integer and Long are two different data types and different objects. So they will be treated as unique elements and not overridden.

Answer: a

Explanation: Set has contains(Object o) method instead of get(Object o) method as get is needed for comparing object and getting corresponding value.

8. What is the difference between TreeSet and SortedSet?

- a) TreeSet is more efficient than SortedSet
- b) SortedSet is more efficient than TreeSet
- c) TreeSet is an interface; SortedSet is a concrete class
- d) SortedSet is an interface; TreeSet is a concrete class

Answer: d

Explanation: SortedSet is an interface. It maintains an ordered set of elements. TreeSet is an implementation of SortedSet.

9. What happens if two threads simultaneously modify TreeSet?

- a) ConcurrentModificationException is thrown
- b) Both threads can perform action successfully
- c) FailFastException is thrown
- d) IteratorModificationException is thrown

Answer: a

Explanation: TreeSet provides fail-fast iterator. Hence when concurrently modifying TreeSet it throws ConcurrentModificationException.

10. What is the unique feature of LinkedHashSet?

- a) It is not a valid class**
- b) It maintains the insertion order and guarantees uniqueness**
- c) It provides a way to store key values with uniqueness**
- d) The elements in the collection are linked to each other**

Answer: b

Explanation: Set is a collection of unique elements. HashSet has the behavior of Set and stores key value pairs. The LinkedHashSet stores the key value pairs in the order of insertion.

1. Which of these standard collection classes implements a linked list data structure?

- a) AbstractList**
- b) LinkedList**
- c) HashSet**
- d) AbstractSet**

Answer: b

Explanation: None.

2. Which of these classes implements Set interface?

- a) ArrayList**
- b) HashSet**
- c) LinkedList**
- d) DynamicList**

Answer: b

Explanation: HashSet and TreeSet implements Set interface where as LinkedList and ArrayList implements List interface.

3. Which of these method is used to add an element to the start of a LinkedList object?

- a) add()**
- b) first()**
- c) AddFirst()**
- d) addFirst()**

Answer: d

Explanation: None.

4. Which of these method of HashSet class is used to add elements to its object?

- a) add()**
- b) Add()**
- c) addFirst()**
- d) insert()**

Answer: a

Explanation: None.

5. Which of these methods can be used to delete the last element in a LinkedList object?

- a) remove()**
- b) delete()**
- c) removeLast()**
- d) deleteLast()**

Answer: c

Explanation: removeLast() and removeFirst() methods are used to remove elements in end and beginning of a linked list.

6. Which of this method is used to change an element in a LinkedList Object?

- a) change()**
- b) set()**

- c) redo()
- d) add()

Answer: b

Explanation: An element in a LinkedList object can be changed by first using get() to obtain the index or location of that object and the passing that location to method set() along with its new value.

7. What will be the output of the following Java code snippet?

```
1.    import java.util.*;
2.
3.    class Linkedlist
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            LinkedList obj = new LinkedList();
10.
11.           obj.add("A");
12.
13.           obj.add("B");
14.
15.           obj.add("C");
16.
17.           obj.addFirst("D");
18.
19.           System.out.println(obj);
20.
21.       }
22.   }
```

- a) [A, B, C]
- b) [D, B, C]
- c) [A, B, C, D]
- d) [D, A, B, C]

Answer: d

Explanation: obj.addFirst("D") method is used to add 'D' to the start of a LinkedList object obj.

Output:

```
$ javac Linkedlist.java
$ java Linkedlist
[D, A, B, C].
```

8. What will be the output of the following Java program?

```
1.    import java.util.*;
2.
3.    class Linkedlist
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            LinkedList obj = new LinkedList();
10.
11.           obj.add("A");
```

```

8.         obj.add("B");
9.         obj.add("C");
10.        obj.removeFirst();
11.        System.out.println(obj);
12.    }
13. }

```

- a) [A, B]
- b) [B, C]
- c) [A, B, C, D]
- d) [A, B, C]

Answer: b

Explanation: None.

Output:

```

$ javac Linkedlist.java
$ java Linkedlist
[B, C]

```

9. What will be the output of the following Java program?

```

1.     import java.util.*;
2.     class Output
3.     {
4.         public static void main(String args[])
5.         {
6.             HashSet obj = new HashSet();
7.             obj.add("A");
8.             obj.add("B");
9.             obj.add("C");
10.            System.out.println(obj + " " + obj.size());
11.        }
12.    }

```

- a) ABC 3
- b) [A, B, C] 3
- c) ABC 2
- d) [A, B, C] 2

Answer: b

Explanation: HashSet obj creates an hash object which implements Set interface, obj.size() gives the number of elements stored in the object obj which in this case is 3.

Output:

```

$ javac Output.java
$ java Output
[A, B, C] 3

```

10. What will be the output of the following Java program?

```
1.    import java.util.*;
2.
3.    class Output
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            TreeSet t = new TreeSet();
10.
11.            t.add("3");
12.
13.            t.add("9");
14.
15.            t.add("1");
16.
17.            t.add("4");
18.
19.            t.add("8");
20.
21.            System.out.println(t);
22.
23.        }
24.    }
```

- a) [1, 3, 5, 8, 9]
- b) [3, 4, 1, 8, 9]
- c) [9, 8, 4, 3, 1]
- d) [1, 3, 4, 8, 9]

Answer: d

Explanation: TreeSet class uses set to store the values added by function add in ascending order using tree for storage
Output:

```
$ javac Output.java
$ java Output
[1, 3, 4, 8, 9].
```

1. Which of these object stores association between keys and values?

- a) Hash table
- b) Map
- c) Array
- d) String

Answer: b

Explanation: None.

2. Which of these classes provide implementation of map interface?

- a) ArrayList
- b) HashMap
- c) LinkedList
- d) DynamicList

Answer: b

Explanation: AbstractMap, WeakHashMap, HashMap and TreeMap provide implementation of map interface.

3. Which of these method is used to remove all keys/values pair from the invoking map?

- a) delete()
- b) remove()
- c) clear()
- d) removeAll()

Answer: b

Explanation: None.

4. Which of these method Map class is used to obtain an element in the map having specified key?

- a) search()
- b) get()
- c) set()
- d) look()

Answer: b

Explanation: None.

5. Which of these methods can be used to obtain set of all keys in a map?

- a) getAll()
- b) getKeys()
- c) keyall()
- d) keySet()

Answer: d

Explanation: keySet() methods is used to get a set containing all the keys used in a map. This method provides set view of the keys in the invoking map.

6. Which of these method is used add an element and corresponding key to a map?

- a) put()
- b) set()
- c) redo()
- d) add()

Answer: a

Explanation: Maps revolve around two basic operations – get() and put(). to put a value into a map, use put(), specifying the key and the value. To obtain a value, call get() , passing the key as an argument. The value is returned.

7. What will be the output of the following Java program?

```
1.    import java.util.*;
2.
3.    class Maps
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            HashMap obj = new HashMap();
10.           obj.put("A", new Integer(1));
11.           obj.put("B", new Integer(2));
12.           obj.put("C", new Integer(3));
13.           System.out.println(obj);
14.       }
```

```
12.      }
```

- a) {A 1, B 1, C 1}
- b) {A, B, C}
- c) {A-1, B-1, C-1}
- d) {A=1, B=2, C=3}

Answer: d

Explanation: None.

Output:

```
$ javac Maps.java
$ java Maps
{A=1, B=2, C=3}
```

8. What will be the output of the following Java program?

```
1.      import java.util.*;
2.
3.      class Maps
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              HashMap obj = new HashMap();
10.             obj.put("A", new Integer(1));
11.             obj.put("B", new Integer(2));
12.             obj.put("C", new Integer(3));
13.             System.out.println(obj.keySet());
14.         }
15.     }
```

- a) [A, B, C]
- b) {A, B, C}
- c) {1, 2, 3}
- d) [1, 2, 3]

Answer: a

Explanation: keySet() method returns a set containing all the keys used in the invoking map. Here keys are characters A, B & C. 1, 2, 3 are the values given to these keys.

Output:

```
$ javac Maps.java
$ java Maps
[A, B, C].
```

9. What will be the output of the following Java program?

```
1.      import java.util.*;
2.
3.      class Maps
4.      {
```

```

4.         public static void main(String args[])
5.         {
6.             HashMap obj = new HashMap();
7.             obj.put("A", new Integer(1));
8.             obj.put("B", new Integer(2));
9.             obj.put("C", new Integer(3));
10.            System.out.println(obj.get("B"));
11.        }
12.    }

```

- a) 1
- b) 2
- c) 3
- d) null

Answer: b

Explanation: obj.get("B") method is used to obtain the value associated with key "B", which is 2.

Output:

```

$ javac Maps.java
$ java Maps
2

```

10. What will be the output of the following Java program?

```

1.     import java.util.*;
2.     class Maps
3.     {
4.         public static void main(String args[])
5.         {
6.             TreeMap obj = new TreeMap();
7.             obj.put("A", new Integer(1));
8.             obj.put("B", new Integer(2));
9.             obj.put("C", new Integer(3));
10.            System.out.println(obj.entrySet());
11.        }
12.    }

```

- a) [A, B, C]
- b) [1, 2, 3]
- c) {A=1, B=2, C=3}
- d) [A=1, B=2, C=3]

Answer: d

Explanation: obj.entrySet() method is used to obtain a set that contains the entries in the map. This method provides set

view of the invoking map.

Output:

```
$ javac Maps.java
$ java Maps
[A=1, B=2, C=3].
```

1. Which of these class object can be used to form a dynamic array?

- a) ArrayList
- b) Map
- c) Vector
- d) ArrayList & Vector

Answer: d

Explanation: Vectors are dynamic arrays, it contains many legacy methods that are not part of collection framework, and hence these methods are not present in ArrayList. But both are used to form dynamic arrays.

2. Which of these are legacy classes?

- a) Stack
- b) Hashtable
- c) Vector
- d) All of the mentioned

Answer: d

Explanation: Stack, Hashtable, Vector, Properties and Dictionary are legacy classes.

3. Which of these is the interface of legacy?

- a) Map
- b) Enumeration
- c) HashMap
- d) Hashtable

Answer: b

Explanation: None.

4. What is the name of a data member of class Vector which is used to store a number of elements in the vector?

- a) length
- b) elements
- c) elementCount
- d) capacity

Answer: c

Explanation: None.

5. Which of these methods is used to add elements in vector at specific location?

- a) add()
- b) set()
- c) AddElement()
- d) addElement()

Answer: d

Explanation: addElement() is used to add data in the vector, to obtain the data we use elementAt() and to first and last element we use firstElement() and lastElement() respectively.

6. What will be the output of the following Java code?

```
1.      import java.util.*;
2.      class vector
```

```

3.      {
4.          public static void main(String args[])
5.          {
6.              Vector obj = new Vector(4,2);
7.              obj.addElement(new Integer(3));
8.              obj.addElement(new Integer(2));
9.              obj.addElement(new Integer(5));
10.             System.out.println(obj.elementAt(1));
11.         }
12.     }

```

- a) 0
- b) 3
- c) 2
- d) 5

Answer: c

Explanation: obj.elementAt(1) returns the value stored at index 1, which is 2.

Output:

```

$ javac vector.java
$ java vector
2

```

7. What will be the output of the following Java code?

```

1.      import java.util.*;
2.      class vector
3.      {
4.          public static void main(String args[])
5.          {
6.              Vector obj = new Vector(4,2);
7.              obj.addElement(new Integer(3));
8.              obj.addElement(new Integer(2));
9.              obj.addElement(new Integer(5));
10.             System.out.println(obj.capacity());
11.         }
12.     }

```

- a) 2
- b) 3
- c) 4
- d) 6

Answer: c

Explanation: None.

Output:

```
$ javac vector.java
$ java vector
4
```

8. What will be the output of the following Java code?

```
1.     import java.util.*;
2.
3.     class vector
4.     {
5.
6.         public static void main(String args[])
7.         {
8.
9.             Vector obj = new Vector(4,2);
10.
11.            obj.addElement(new Integer(3));
12.
13.            obj.addElement(new Integer(2));
14.
15.            obj.addElement(new Integer(6));
16.
17.            obj.insertElementAt(new Integer(8), 2);
18.
19.            System.out.println(obj);
20.
21.        }
22.    }
```

a) [3, 2, 6]

b) [3, 2, 8]

c) [3, 2, 6, 8]

d) [3, 2, 8, 6]

Answer: d

Explanation: None.

Output:

```
$ javac vector.java
$ java vector
[3, 2, 8, 6].
```

9. What will be the output of the following Java code?

```
1.     import java.util.*;
2.
3.     class vector
4.     {
5.
6.         public static void main(String args[])
7.         {
8.
9.             Vector obj = new Vector(4,2);
10.
11.            obj.addElement(new Integer(3));
```

```

8.         obj.addElement(new Integer(2));
9.         obj.addElement(new Integer(5));
10.        obj.removeAll(obj);
11.        System.out.println(obj.isEmpty());
12.    }
13. }

```

- a) 0
- b) 1
- c) true
- d) false

Answer: c

Explanation: firstly elements 3, 2, 5 are entered in the vector obj, but when obj.removeAll(obj); is executed all the elements are deleted and vector is empty, hence obj.isEmpty() returns true.

Output:

```

$ javac vector.java
$ java vector
true

```

10. What will be the output of the following Java code?

```

1.     import java.util.*;
2.     class stack
3.     {
4.         public static void main(String args[])
5.         {
6.             Stack obj = new Stack();
7.             obj.push(new Integer(3));
8.             obj.push(new Integer(2));
9.             obj.pop();
10.            obj.push(new Integer(5));
11.            System.out.println(obj);
12.        }
13.    }

```

- a) [3, 5]
- b) [3, 2]
- c) [3, 2, 5]
- d) [3, 5, 2]

Answer: a

Explanation: push() and pop() are standard functions of the class stack, push() inserts in the stack and pop removes from the stack. 3 & 2 are inserted using push() the pop() is used which removes 2 from the stack then again push is used to insert 5 hence stack contains elements 3 & 5.

Output:

```
$ javac stack.java
$ java stack
[3, 5].
```

1. Which of these class object uses the key to store value?

- a) Dictionary**
- b) Map**
- c) Hashtable**
- d) All of the mentioned**

Answer: d

Explanation: Dictionary, Map & Hashtable all implement Map interface hence all of them uses keys to store value in the object.

2. Which of these method is used to insert value and its key?

- a) put()**
- b) set()**
- c) insertElement()**
- d) addElement()**

Answer: a

Explanation: None.

3. Which of these is the interface of legacy is implemented by Hashtable and Dictionary classes?

- a) Map**
- b) Enumeration**
- c) HashMap**
- d) Hashtable**

Answer: a

Explanation: Dictionary, Map & Hashtable all implement Map interface hence all of them uses keys to store value in the object.

4. Which of these is a class which uses String as a key to store the value in object?

- a) Array**
- b) ArrayList**
- c) Dictionary**
- d) Properties**

Answer: d

Explanation: None.

5. Which of these methods is used to retrieve the elements in properties object at specific location?

- a) get()**
- b) Elementat()**
- c) ElementAt()**
- d) getProperty()**

Answer: d

Explanation: None.

6. What will be the output of the following Java code?

```
1.      import java.util.*;

2.      class hashtable
```

```

3.      {
4.          public static void main(String args[])
5.          {
6.              Hashtable obj = new Hashtable();
7.              obj.put("A", new Integer(3));
8.              obj.put("B", new Integer(2));
9.              obj.put("C", new Integer(8));
10.             System.out.print(obj.contains(new Integer(5)));
11.         }
12.     }

```

- a) 0
- b) 1
- c) true
- d) false

Answer: d

Explanation: Hashtable object obj contains values 3, 2, 8 when obj.contains(new Integer(5)) is executed it searches for 5 in the hashtable since it is not present false is returned.

Output:

```

$ javac hashtable.java
$ java hashtable
false

```

7. What will be the output of the following Java code?

```

1.      import java.util.*;
2.
3.      class hashtable
4.      {
5.          public static void main(String args[])
6.          {
7.              Hashtable obj = new Hashtable();
8.              obj.put("A", new Integer(3));
9.              obj.put("B", new Integer(2));
10.             obj.put("C", new Integer(8));
11.             obj.clear();
12.             System.out.print(obj.size());
13.         }

```

- a) 0
- b) 1
- c) 2

d) 3

Answer: a

Explanation: None.

Output:

```
$ javac hashtable.java
$ java hashtable
0
```

8. What will be the output of the following Java code?

```
1.      import java.util.*;
2.
3.      class hashtable
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              Hashtable obj = new Hashtable();
10.
11.             obj.put("A", new Integer(3));
12.
13.             obj.put("B", new Integer(2));
14.
15.             obj.put("C", new Integer(8));
16.
17.             obj.remove(new String("A"));
18.
19.             System.out.print(obj);
20.
21.         }
22.     }
```

a) {C=8, B=2}

b) [C=8, B=2]

c) {A=3, C=8, B=2}

d) [A=3, C=8, B=2]

Answer: a

Explanation: None.

Output:

```
$ javac hashtable.java
$ java hashtable
{C=8, B=2}
```

9. What will be the output of the following Java code?

```
1.      import java.util.*;
2.
3.      class hashtable
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              Hashtable obj = new Hashtable();
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.
48.
49.
50.
51.
52.
53.
54.
55.
56.
57.
58.
59.
60.
61.
62.
63.
64.
65.
66.
67.
68.
69.
70.
71.
72.
73.
74.
75.
76.
77.
78.
79.
80.
81.
82.
83.
84.
85.
86.
87.
88.
89.
90.
91.
92.
93.
94.
95.
96.
97.
98.
99.
100.
```

```

7.         obj.put("A", new Integer(3));
8.         obj.put("B", new Integer(2));
9.         obj.put("C", new Integer(8));
10.        System.out.print(obj.toString());
11.    }
12. }

```

- a) {C=8, B=2}
- b) [C=8, B=2]
- c) {A=3, C=8, B=2}
- d) [A=3, C=8, B=2]

Answer: c

Explanation: obj.toString returns String equivalent of the hashtable, which can also be obtained by simply writing System.out.print(obj); as print system automatically converts the obj toString equivalent.

Output:

```

$ javac hashtable.java
$ java hashtable
{A=3, C=8, B=2}

```

10. What will be the output of the following Java code?

```

1.     import java.util.*;
2.     class properties
3.     {
4.         public static void main(String args[])
5.         {
6.             Properties obj = new Properties();
7.             obj.put("AB", new Integer(3));
8.             obj.put("BC", new Integer(2));
9.             obj.put("CD", new Integer(8));
10.            System.out.print(obj.keySet());
11.        }
12.    }

```

- a) {AB, BC, CD}
- b) [AB, BC, CD]
- c) [3, 2, 8]
- d) {3, 2, 8}

Answer: b

Explanation: obj.keySet() returns a set containing all the keys used in properties object, here obj contains keys AB, BC, CD therefore obj.keySet() returns [AB, BC, CD].

Output:

```

$ javac properties.java
$ java properties

```


1. Which of these class object has an architecture similar to that of array?

- a) Bitset**
- b) Map**
- c) Hashtable**
- d) All of the mentioned**

Answer: a

Explanation: Bitset class creates a special type of array that holds bit values. This array can increase in size as needed.

2. Which of these method is used to make a bit zero specified by the index?

- a) put()**
- b) set()**
- c) remove()**
- d) clear()**

Answer: d

Explanation: None.

3. Which of these method is used to calculate number of bits required to hold the BitSet object?

- a) size()**
- b) length()**
- c) indexes()**
- d) numberOfBits()**

Answer: b

Explanation: None.

4. Which of these is a method of class Date which is used to search whether object contains a date before the specified date?

- a) after()**
- b) contains()**
- c) before()**
- d) compareTo()**

Answer: c

Explanation: before() returns true if the invoking Date object contains a date that is earlier than one specified by date, otherwise it returns false.

5. Which of these methods is used to retrieve elements in BitSet object at specific location?

- a) get()**
- b) Elementat()**
- c) ElementAt()**
- d) getProperty()**

Answer: a

Explanation: None.

6. What will be the output of the following Java code?

```
1.      import java.util.*;
2.
3.      class Bitset
4.      {
5.          public static void main(String args[])
6.          {
```

```

6.         BitSet obj = new BitSet(5);
7.         for (int i = 0; i < 5; ++i)
8.             obj.set(i);
9.         obj.clear(2);
10.        System.out.print(obj);
11.    }
12. }

```

- a) {0, 1, 3, 4}
- b) {0, 1, 2, 4}
- c) {0, 1, 2, 3, 4}
- d) {0, 0, 0, 3, 4}

Answer: a

Explanation: None.

Output:

```

$ javac Bitset.java
$ java Bitset
{0, 1, 3, 4}

```

7. What will be the output of the following Java code?

```

1.     import java.util.*;
2.     class Bitset
3.     {
4.         public static void main(String args[])
5.         {
6.             BitSet obj = new BitSet(5);
7.             for (int i = 0; i < 5; ++i)
8.                 obj.set(i);
9.             obj.clear(2);
10.            System.out.print(obj.length() + " " + obj.size());
11.        }
12.    }

```

- a) 4 64
- b) 5 64
- c) 5 128
- d) 4 128

Answer: b

*Explanation: obj.length() returns the length allotted to object obj at time of initialization and obj.size() returns the size of current object obj, each BitSet element is given 16 bits therefore the size is $4 * 16 = 64$, whereas length is still 5.*

Output:

```

$ javac Bitset.java

```

```
$ java Bitset
5 64
```

8. What will be the output of the following Java code?

```
1.    import java.util.*;
2.
3.    class Bitset
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            BitSet obj = new BitSet(5);
10.           for (int i = 0; i < 5; ++i)
11.
12.               obj.set(i);
13.
14.           System.out.print(obj.get(3));
15.
16.       }
17.
18.   }
```

- a) 2
- b) 3
- c) 4
- d) 5

Answer: a

Explanation: None.

Output:

```
$ javac Bitset.java
$ java Bitset
2
```

9. What will be the output of the following Java code?

```
1.    import java.util.*;
2.
3.    class date
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            Date obj = new Date();
10.
11.            System.out.print(obj);
12.
13.        }
14.
15.    }
```

- a) Prints Present Date
- b) Runtime Error
- c) Any Garbage Value
- d) Prints Present Time & Date

Answer: d

Explanation: None.

Output:

```
$ javac date.java
$ java date
Tue Jun 11 11:29:57 PDT 2013
```

10. What will be the output of the following Java code?

```
1.      import java.util.*;
2.
3.      class Bitset
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              BitSet obj1 = new BitSet(5);
10.             BitSet obj2 = new BitSet(10);
11.             for (int i = 0; i < 5; ++i)
12.                 obj1.set(i);
13.             for (int i = 3; i < 13; ++i)
14.                 obj2.set(i);
15.             obj1.and(obj2);
16.             System.out.print(obj1);
17.         }
18.     }
```

- a) {0, 1}**
- b) {2, 4}**
- c) {3, 4}**
- d) {3, 4, 5}**

Answer: c

Explanation: obj1.and(obj2) returns an BitSet object which contains elements common to both the object obj1 and obj2 and stores this BitSet in invoking object that is obj1. Hence obj1 contains 3 & 4.

Output:

```
$ javac Bitset.java
$ java Bitset
{3, 4}
```

1. What is Remote method invocation (RMI)?

- a) RMI allows us to invoke a method of java object that executes on another machine**
- b) RMI allows us to invoke a method of java object that executes on another Thread in multithreaded programming**
- c) RMI allows us to invoke a method of java object that executes parallely in same machine**
- d) None of the mentioned**

Answer: a

Explanation: Remote method invocation RMI allows us to invoke a method of java object that executes on another machine.

2. Which of these package is used for remote method invocation?

- a) java.applet
- b) java.rmi
- c) java.lang.rmi
- d) java.lang.reflect

Answer: b

Explanation: None.

3. Which of these methods are member of Remote class?

- a) checkIP()
- b) addLocation()
- c) AddServer()
- d) None of the mentioned

Answer: d

Explanation: Remote class does not define any methods, its purpose is simply to indicate that an interface uses remote methods.

4. Which of these Exceptions is thrown by remote method?

- a) RemoteException
- b) InputOutputException
- c) RemoteAccessException
- d) RemoteInputOutputException

Answer: a

Explanation: All remote methods throw RemoteException.

5. Which of these class is used for creating a client for a server-client operations?

- a) serverClientjava
- b) Client.java
- c) AddClient.java
- d) ServerClient.java

Answer: c

Explanation: None.

6. Which of these package is used for all the text related modifications?

- a) java.text
- b) java.awt
- c) java.lang.text
- d) java.text.modify

Answer: a

Explanation: java.text provides capabilities for formatting, searching and manipulating text.

7. What will be the output of the following Java code?

```
1.      import java.lang.reflect.*;
2.
3.      class Additional_packages
4.      {
5.          public static void main(String args[])
6.          {
7.              try
```

```

7.         {
8.             Class c = Class.forName("java.awt.Dimension");
9.             Constructor constructors[] = c.getConstructors();
10.            for (int i = 0; i < constructors.length; i++)
11.                System.out.println(constructors[i]);
12.        }
13.    catch (Exception e)
14.    {
15.        System.out.print("Exception");
16.    }
17. }
18. }
```

- a) Program prints all the constructors of 'java.awt.Dimension' package
- b) Program prints all the possible constructors of class 'Class'
- c) Program prints "Exception"
- d) Runtime Error

Answer: a

Explanation: None.

Output:

```

$ javac Additional_packages.java
$ java Additional_packages
public java.awt.Dimension(java.awt.Dimension)
public java.awt.Dimension()
public java.awt.Dimension(int,int)
```

8. What will be the output of the following Java code?

```

1.    import java.lang.reflect.*;
2.
3.    class Additional_packages
4.    {
5.        public static void main(String args[])
6.        {
7.            try
8.            {
9.                Class c = Class.forName("java.awt.Dimension");
10.               Field fields[] = c.getFields();
11.               for (int i = 0; i < fields.length; i++)
12.                   System.out.println(fields[i]);
13.            }
14.        }
```

```

13.         catch (Exception e)
14.         {
15.             System.out.print("Exception");
16.         }
17.     }
18. }

```

- a) Program prints all the constructors of ‘java.awt.Dimension’ package
- b) Program prints all the methods of ‘java.awt.Dimension’ package
- c) Program prints all the data members of ‘java.awt.Dimension’ package
- d) program prints all the methods and data member of ‘java.awt.Dimension’ package

Answer: c

Explanation: None.

Output:

```

$ javac Additional_packages.java
$ java Additional_packages
public int java.awt.Dimension.width
public int java.awt.Dimension.height

```

9. What is the length of the application box made in the following Java program?

```

1.     import java.awt.*;
2.     import java.applet.*;
3.     public class myapplet extends Applet
4.     {
5.         Graphic g;
6.         g.drawString("A Simple Applet",20,20);
7.     }

```

- a) 20
- b) Default value
- c) Compilation Error
- d) Runtime Error

Answer: c

Explanation: To implement the method drawString we need first need to define abstract method of AWT that is paint() method. Without paint() method we cannot define and use drawString or any Graphic class methods.

10. What will be the output of the following Java program?

```

1.     import java.lang.reflect.*;
2.     class Additional_packages
3.     {
4.         public static void main(String args[])
5.         {
6.             try

```

```

7.         {
8.             Class c = Class.forName("java.awt.Dimension");
9.             Method methods[] = c.getMethods();
10.            for (int i = 0; i < methods.length; i++)
11.                System.out.println(methods[i]);
12.        }
13.        catch (Exception e)
14.        {
15.            System.out.print("Exception");
16.        }
17.    }
18. }

```

- a) Program prints all the constructors of 'java.awt.Dimension' package
- b) Program prints all the methods of 'java.awt.Dimension' package
- c) Program prints all the data members of 'java.awt.Dimension' package
- d) program prints all the methods and data member of 'java.awt.Dimension' package

Answer: b

Explanation: None.

Output:

```

$ javac Additional_packages.java
$ java Additional_packages
public int java.awt.Dimension.hashCode()
public boolean java.awt.Dimension.equals(java.lang.Object)
public java.lang.String java.awt.Dimension.toString()
public java.awt.Dimension java.awt.Dimension.getSize()
public void java.awt.Dimension.setSize(double,double)
public void java.awt.Dimension.setSize(int,int)
public void java.awt.Dimension.setSize(java.awt.Dimension)
public double java.awt.Dimension.getHeight()
public double java.awt.Dimension.getWidth()
public java.lang.Object java.awt.geom.Dimension2D.clone()
public void java.awt.geom.Dimension2D.setSize(java.awt.geom.Dimension2D)
public final native java.lang.Class java.lang.Object.getClass()
public final native void java.lang.Object.notify()
public final native void java.lang.Object.notifyAll()
public final native void java.lang.Object.wait(long)
public final void java.lang.Object.wait(long,int)
public final void java.lang.Object.wait()

```

1. Which of these packages contain all the collection classes?

- a) java.lang
- b) java.util
- c) java.net
- d) java.awt

Answer: b

Explanation: None.

2. Which of these classes is not part of Java's collection framework?

- a) Maps

- b) Array
- c) Stack
- d) Queue

Answer: a

Explanation: Maps is not a part of collection framework.

3. Which of this interface is not a part of Java's collection framework?

- a) List
- b) Set
- c) SortedMap
- d) SortedList

Answer: d

Explanation: SortedList is not a part of collection framework.

4. Which of these methods deletes all the elements from invoking collection?

- a) clear()
- b) reset()
- c) delete()
- d) refresh()

Answer: a

Explanation: clear() method removes all the elements from invoking collection.

5. What is Collection in Java?

- a) A group of objects
- b) A group of classes
- c) A group of interfaces
- d) None of the mentioned

Answer: a

Explanation: A collection is a group of objects, it is similar to String Template Library (STL) of C++ programming language.

6. What will be the output of the following Java program?

```
1.      import java.util.*;
2.
3.      class Array
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              int array[] = new int [5];
10.
11.             for (int i = 5; i > 0; i--)
12.
13.                 array[5-i] = i;
14.
15.             Arrays.fill(array, 1, 4, 8);
16.
17.             for (int i = 0; i < 5 ; i++)
18.
19.                 System.out.print(array[i]);
20.
21.             }
22.
23.     }
```

- a) 12885
- b) 12845
- c) 58881
- d) 54881

Answer: c

Explanation: array was containing 5,4,3,2,1 but when method Arrays.fill(array, 1, 4, 8) is called it fills the index location starting with 1 to 4 by value 8 hence array becomes 5,8,8,8,1.

Output:

```
$ javac Array.java
$ java Array
58881
```

7. What will be the output of the following Java program?

```
1.    import java.util.*;
2.
3.    class Bitset
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            BitSet obj = new BitSet(5);
10.
11.           for (int i = 0; i < 5; ++i)
12.
13.               obj.set(i);
14.
15.           obj.clear(2);
16.
17.           System.out.print(obj);
18.
19.       }
20.
21.   }
```

- a) {0, 1, 3, 4}
- b) {0, 1, 2, 4}
- c) {0, 1, 2, 3, 4}
- d) {0, 0, 0, 3, 4}

Answer: a

Explanation: None.

Output:

```
$ javac Bitset.java
$ java Bitset
{0, 1, 3, 4}
```

1. Which of these return type of hasNext() method of an iterator?

- a) Integer
- b) Double
- c) Boolean
- d) Collections Object

Answer: c

Explanation: hasNext() returns boolean values true or false.

2. Which of these methods is used to obtain an iterator to the start of collection?

- a) start()
- b) begin()
- c) iteratorSet()
- d) iterator()

Answer: d

Explanation: To obtain an iterator to the start of the start of the collection we use iterator() method.

3. Which of these methods can be used to move to next element in a collection?

- a) next()
- b) move()
- c) shuffle()
- d) hasNext()

Answer: a

Explanation: None.

4. Which of these iterators can be used only with List?

- a) Setiterator
- b) ListIterator
- c) Literator
- d) None of the mentioned

Answer: b

Explanation: None.

5. Which of these is a method of ListIterator used to obtain index of previous element?

- a) previous()
- b) previousIndex()
- c) back()
- d) goBack()

Answer: b

Explanation: previousIndex() returns index of previous element. if there is no previous element then -1 is returned.

6. Which of these exceptions is thrown by remover() method?

- a) IOException
- b) SystemException
- c) ObjectNotFoundException
- d) IllegalStateException

Answer: d

Explanation: None.

7. What will be the output of the following Java program?

```
1.    import java.util.*;
2.
3.    class Collection_iterators
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            ListIterator a = list.listIterator();
10.
11.            if(a.previousIndex() != -1)
```

```

8.         while(a.hasNext())
9.             System.out.print(a.next() + " ");
10.        else
11.            System.out.print("EMPTY");
12.    }
13. }

```

- a) 0
- b) 1
- c) -1
- d) EMPTY

Answer: d

Explanation: None.

Output:

```

$ javac Collection_iterators.java
$ java Collection_iterators
EMPTY

```

8. What will be the output of the following Java program?

```

1.  import java.util.*;
2.  class Collection_iterators
3.  {
4.      public static void main(String args[])
5.      {
6.          LinkedList list = new LinkedList();
7.          list.add(new Integer(2));
8.          list.add(new Integer(8));
9.          list.add(new Integer(5));
10.         list.add(new Integer(1));
11.         Iterator i = list.iterator();
12.         Collections.reverse(list);
13.         while(i.hasNext())
14.             System.out.print(i.next() + " ");
15.     }
16. }

```

- a) 2 8 5 1
- b) 1 5 8 2
- c) 2
- d) 2 1 8 5

Answer: b

Explanation: `Collections.reverse(list)` reverses the given list, the list was 2->8->5->1 after reversing it became 1->5->8->2.

Output:

```
$ javac Collection_iterators.java
$ java Collection_iterators
1 5 8 2
```

9. What will be the output of the following Java program?

```
1.    import java.util.*;
2.
3.    class Collection_iterators
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            LinkedList list = new LinkedList();
10.
11.            list.add(new Integer(2));
12.
13.            list.add(new Integer(8));
14.
15.            list.add(new Integer(5));
16.
17.            list.add(new Integer(1));
18.
19.            Iterator i = list.iterator();
20.
21.            Collections.reverse(list);
22.
23.            Collections.sort(list);
24.
25.            while(i.hasNext())
26.
27.                System.out.print(i.next() + " ");
28.
29.        }
30.    }
```

a) 2 8 5 1

b) 1 5 8 2

c) 1 2 5 8

d) 2 1 8 5

Answer: c

Explanation: `Collections.sort(list)` sorts the given list, the list was 2->8->5->1 after sorting it became 1->2->5->8.

Output:

```
$ javac Collection_iterators.java
$ java Collection_iterators
1 2 5 8
```

10. What will be the output of the following Java program?

```
1.    import java.util.*;
2.
3.    class Collection_iterators
```

```

3.      {
4.          public static void main(String args[])
5.      {
6.          LinkedList list = new LinkedList();
7.          list.add(new Integer(2));
8.          list.add(new Integer(8));
9.          list.add(new Integer(5));
10.         list.add(new Integer(1));
11.         Iterator i = list.iterator();
12.         Collections.reverse(list);
13.         Collections.shuffle(list);
14.         i.next();
15.         i.remove();
16.         while(i.hasNext())
17.             System.out.print(i.next() + " ");
18.     }
19. }

```

- a) 2 8 5
- b) 2 1 8
- c) 2 5 8
- d) 8 5 1

Answer: b

Explanation: i.next() returns the next element in the iteration. i.remove() removes from the underlying collection the last element returned by this iterator (optional operation). This method can be called only once per call to next(). The behavior of an iterator is unspecified if the underlying collection is modified while the iteration is in progress in any way other than by calling this method.

Output:

```

$ javac Collection_iterators.java
$ java Collection_iterators
2 1 8

```

(output will be different on your system)

1. Which of the below is not a subinterface of Queue?

- a) BlockingQueue
- b) BlockingEnque
- c) TransferQueue
- d) BlockingQueue

Answer: b

Explanation: BlockingQueue, TransferQueue and BlockingQueue are subinterfaces of Queue.

2. What is the remaining capacity of BlockingQueue whose intrinsic capacity is not defined?

- a) Integer.MAX_VALUE

- b) `BigDecimal.MAX_VALUE`
- c) `999999999`
- d) `Integer.INFINITY`

Answer: a

Explanation: A `BlockingQueue` without any intrinsic capacity constraints always reports a remaining capacity of `Integer.MAX_VALUE`.

Answer: a
Explanation: `PriorityQueue` is not synchronized. `BlockingPriorityQueue` is the thread safe implementation.

4. What is difference between `dequeue()` and `peek()` function of java?
- a) `dequeue()` and `peek()` remove and return the next time in line
 - b) `dequeue()` and `peek()` return the next item in line
 - c) `dequeue()` removes and returns the next item in line while `peek()` returns the next item in line
 - d) `peek()` removes and returns the next item in line while `dequeue()` returns the next item in line

Answer: c
Explanation: `dequeue()` removes the item next in line. `peek()` returns the item without removing it from the queue.

5. What is the difference between Queue and Stack?
- a) Stack is LIFO; Queue is FIFO
 - b) Queue is LIFO; Stack is FIFO
 - c) Stack and Queue is FIFO
 - d) Stack and Queue is LIFO

Answer: a
Explanation: Stack is Last in First out (LIFO) and Queue is First in First out (FIFO).

6. What are the use of front and rear pointers in CircularQueue implementation?
- a) Front pointer points to first element; rear pointer points to the last element
 - b) Rear pointer points to first element; front pointer points to the last element
 - c) Front and read pointers point to the first element
 - d) Front pointer points to the first element; rear pointer points to null object

Answer: c
Explanation: CircularQueue implementation is an abstract class where first and rear pointer point to the same object.

7. What is the correct method used to insert and delete items from the queue?
- a) push and pop
 - b) enqueue and dequeue
 - c) enqueue and peek
 - d) add and remove

Answer: b
Explanation: enqueue is pushing item into queue; dequeue is removing item from queue; peek returns object without removing it from queue.
Stack uses push and pop methods. add and remove are used in the list.

8. Which data structure is used in Breadth First Traversal of a graph?
- a) Stack
 - b) Queue
 - c) Array
 - d) Tree

Answer: b
Explanation: In Breadth First Traversal of graph the nodes at the same level are accessed in the order of retrieval (i.e FIFO).

9. Where does a new element be inserted in linked list implementation of a queue?

- a) Head of list
- b) Tail of list
- c) At the centre of list
- d) All the old entries are pushed and then the new element is inserted

Answer: b

Explanation: To maintain FIFO, newer elements are inserted to the tail of the list.

10. If the size of the array used to implement a circular queue is MAX_SIZE. How rear moves to traverse inorder to insert an element in the queue?

- a) $\text{rear} = (\text{rear} \% 1) + \text{MAX_SIZE}$
- b) $\text{rear} = (\text{rear} + 1) \% \text{MAX_SIZE}$
- c) $\text{rear} = \text{rear} + (1 \% \text{MAX_SIZE})$
- d) $\text{rear} = \text{rear} \% (\text{MAX_SIZE} + 1)$

Answer: b

Explanation: The front and rear pointer of circular queue point to the first element.

1. Which of these standard collection classes implements all the standard functions on list data structure?

- a) Array
- b) LinkedList
- c) HashSet
- d) AbstractSet

Answer: a

Explanation: None.

2. Which of this method is used to make all elements of an equal to specified value?

- a) add()
- b) fill()
- c) all()
- d) set()

Answer: b

Explanation: fill() method assigns a value to all the elements in an array, in other words, it fills the array with specified value.

3. Which of these method of Array class is used sort an array or its subset?

- a) binarysort()
- b) bubblesort()
- c) sort()
- d) insert()

Answer: c

Explanation: None.

4. Which of these methods can be used to search an element in a list?

- a) find()
- b) sort()
- c) get()
- d) binarysearch()

Answer: d

Explanation: binarysearch() method uses binary search to find a specified value. This method must be applied to sorted arrays.

5. What will be the output of the following Java program?


```

1.     import java.util.*;
2.
3.     class ArrayList
4.     {
5.
6.         public static void main(String args[])
7.         {
8.
9.             ArrayList obj1 = new ArrayList();
10.            ArrayList obj2 = new ArrayList();
11.
12.            obj1.add("A");
13.            obj1.add("B");
14.            obj2.add("A");
15.            obj2.add(1, "B");
16.
17.            System.out.println(obj1.equals(obj2));
18.
19.        }
20.    }

```

- a) 0
- b) 1
- c) true
- d) false

Answer: c

Explanation: obj1 and obj2 are an object of class ArrayList hence it is a dynamic array which can increase and decrease its size. obj.add("X") adds to the array element X and obj.add(1, "X") adds element x at index position 1 in the list, Both the objects obj1 and obj2 contain same elements i.e A & B thus obj1.equals(obj2) method returns true.

Output:

```

$ javac ArrayList.java
$ java ArrayList
true

```

6. What will be the output of the following Java program?

```

1.     import java.util.*;
2.
3.     class Array
4.     {
5.
6.         public static void main(String args[])
7.         {
8.
9.             int array[] = new int [5];
10.            for (int i = 5; i > 0; i--)
11.
12.                array[5 - i] = i;
13.
14.            Arrays.sort(array);
15.
16.            for (int i = 0; i < 5; ++i)

```

```
11.         System.out.print(array[i]);  
12.     }  
13. }
```

- a) 12345
- b) 54321
- c) 1234
- d) 5432

Answer: a

Explanation: Arrays.sort(array) method sorts the array into 1,2,3,4,5.

Output:

```
$ javac Array.java  
$ java Array  
12345
```

7. What will be the output of the following Java program?

```
1.     import java.util.*;  
2.     class Array  
3.     {  
4.         public static void main(String args[])  
5.         {  
6.             int array[] = new int [5];  
7.             for (int i = 5; i > 0; i--)  
8.                 array[5 - i] = i;  
9.             Arrays.sort(array);  
10.            System.out.print(Arrays.binarySearch(array, 4));  
11.        }  
12.    }
```

- a) 2
- b) 3
- c) 4
- d) 5

Answer: b

Explanation: None.

Output:

```
$ javac Array.java  
$ java Array  
3
```

1. Which of these interface declares core method that all collections will have?

- a) set
- b) EventListner
- c) Comparator
- d) Collection

Answer: d
Explanation: Collection interfaces defines core methods that all the collections like set, map, arrays etc will have.

2. Which of these interface handle sequences?

- a) Set**
- b) List**
- c) Comparator**
- d) Collection**

Answer: b
Explanation: None.

3. Which of this interface must contain a unique element?

- a) Set**
- b) List**
- c) Array**
- d) Collection**

Answer: a
Explanation: Set interface extends collection interface to handle sets, which must contain unique elements.

4. Which of these is a Basic interface that all other interface inherits?

- a) Set**
- b) Array**
- c) List**
- d) Collection**

Answer: d
Explanation: Collection interface is inherited by all other interfaces like Set, Array, Map etc. It defines core methods that all the collections like set, map, arrays etc will have

5. Which of these is static variable defined in Collections?

- a) EMPTY_SET**
- b) EMPTY_LIST**
- c) EMPTY_MAP**
- d) All of the mentioned**

Answer: d
Explanation: None.

6. What will be the output of the following Java program?

```
1.      import java.util.*;
2.      class Array
3.      {
4.          public static void main(String args[])
5.          {
6.              int array[] = new int [5];
7.              for (int i = 5; i > 0; i--)
8.                  array[5 - i] = i;
9.              Arrays.sort(array);
10.         for (int i = 0; i < 5; ++i)
```

```
11.         System.out.print(array[i]);  
12.     }  
13. }
```

- a) 12345
- b) 54321
- c) 1234
- d) 5432

Answer: a

Explanation: Arrays.sort(array) method sorts the array into 1,2,3,4,5.

Output:

```
$ javac Array.java  
$ java Array  
12345
```

7. What will be the output of the following Java program?

```
1.     import java.util.*;  
2.     class Collection_Algos  
3.     {  
4.         public static void main(String args[])  
5.         {  
6.             LinkedList list = new LinkedList();  
7.             list.add(new Integer(2));  
8.             list.add(new Integer(8));  
9.             list.add(new Integer(5));  
10.            list.add(new Integer(1));  
11.            Iterator i = list.iterator();  
12.            Collections.reverse(list);  
13.            Collections.sort(list);  
14.            while(i.hasNext())  
15.                System.out.print(i.next() + " ");  
16.        }  
17.    }
```

- a) 2 8 5 1
- b) 1 5 8 2
- c) 1 2 5 8
- d) 2 1 8 5

Answer: c

Explanation: Collections.sort(list) sorts the given list, the list was 2->8->5->1 after sorting it became 1->2->5->8.

Output:

```
$ javac Collection_Algos.java
$ java Collection_Algos
1 2 5 8
```

8. What will be the output of the following Java program?

```
1.      import java.util.*;
2.
3.      class Collection_Algos
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              LinkedList list = new LinkedList();
10.
11.              list.add(new Integer(2));
12.
13.              list.add(new Integer(8));
14.
15.              list.add(new Integer(5));
16.
17.              list.add(new Integer(1));
18.
19.              Iterator i = list.iterator();
20.
21.              Collections.reverse(list);
22.
23.              Collections.shuffle(list);
24.
25.              while(i.hasNext())
26.
27.                  System.out.print(i.next() + " ");
28.
29.          }
30.
31.      }
```

- a) 2 8 5 1
- b) 1 5 8 2
- c) 1 2 5 8
- d) Any random order

Answer: d

Explanation: shuffle – randomizes all the elements in a list.

Output:

```
$ javac Collection_Algos.java
$ java Collection_Algos
1 5 2 8
```

(output will be different on your system)

1. Which of these is an incorrect form of using method max() to obtain a maximum element?

- a) max(Collection c)
- b) max(Collection c, Comparator comp)
- c) max(Comparator comp)
- d) max(List c)

Answer: c

Explanation: Its illegal to call max() only with comparator, we need to give the collection to be searched into.

2. Which of these methods sets every element of a List to a specified object?

- a) set()
- b) fill()
- c) Complete()
- d) add()

Answer: b

Explanation: None.

3. Which of these methods can randomize all elements in a list?

- a) rand()
- b) randomize()
- c) shuffle()
- d) ambiguous()

Answer: c

Explanation: shuffle – randomizes all the elements in a list.

4. Which of these methods can convert an object into a List?

- a) SetList()
- b) ConvertList()
- c) singletonList()
- d) CopyList()

Answer: c

Explanation: singletonList() returns the object as an immutable List. This is an easy way to convert a single object into a list. This was added by Java 2.0.

5. Which of these is true about unmodifiableCollection() method?

- a) unmodifiableCollection() returns a collection that cannot be modified
- b) unmodifiableCollection() method is available only for List and Set
- c) unmodifiableCollection() is defined in Collection class
- d) none of the mentioned

Answer: b

Explanation: unmodifiableCollection() is available for al collections, Set, Map, List etc.

6. What will be the output of the following Java program?

```
1.      import java.util.*;
2.
3.      class Collection_Algos
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              LinkedList list = new LinkedList();
10.
11.              list.add(new Integer(2));
12.
13.              list.add(new Integer(8));
14.
15.              list.add(new Integer(5));
16.
17.              list.add(new Integer(1));
18.
19.              Iterator i = list.iterator();
```

```

12.         while(i.hasNext())
13.             System.out.print(i.next() + " ");
14.     }
15. }

```

- a) 2 8 5 1
- b) 1 5 8 2
- c) 2
- d) 2 1 8 5

Answer: a

Explanation: None.

Output:

```

$ javac Collection_Algos.java
$ java Collection_Algos
2 8 5 1

```

7. What will be the output of the following Java program?

```

1.     import java.util.*;
2.
3.     class Collection_Algos
4.     {
5.
6.         public static void main(String args[])
7.         {
8.
9.             LinkedList list = new LinkedList();
10.
11.             list.add(new Integer(2));
12.             list.add(new Integer(8));
13.             list.add(new Integer(5));
14.             list.add(new Integer(1));
15.
16.             Iterator i = list.iterator();
17.             Collections.reverse(list);
18.
19.             while(i.hasNext())
20.                 System.out.print(i.next() + " ");
21.         }
22.     }

```

- a) 2 8 5 1
- b) 1 5 8 2
- c) 2
- d) 2 1 8 5

Answer: b

Explanation: Collections.reverse(list) reverses the given list, the list was 2->8->5->1 after reversing it became 1->5->8->2.

Output:

```
$ javac Collection_Algos.java
$ java Collection_Algos
1 5 8 2
```

1. When does Exceptions in Java arises in code sequence?

- a) Run Time
- b) Compilation Time
- c) Can Occur Any Time
- d) None of the mentioned

Answer: a

Explanation: Exceptions in Java are run-time errors.

2. Which of these keywords is not a part of exception handling?

- a) try
- b) finally
- c) thrown
- d) catch

Answer: c

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

3. Which of these keywords must be used to monitor for exceptions?

- a) try
- b) finally
- c) throw
- d) catch

Answer: a

Explanation: None.

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

- a) try
- b) finally
- c) throw
- d) catch

Answer: d

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

5. Which of these keywords is used to manually throw an exception?

- a) try
- b) finally
- c) throw
- d) catch

Answer: c

Explanation: None.

6. What will be the output of the following Java program?

```
1.      class exception_handling
2.      {
3.          public static void main(String args[])
4.          {
5.              try
```



```
6.         {
7.             System.out.print("Hello" + " " + 1 / 0);
8.         }
9.         catch(ArithmeticException e)
10.        {
11.            System.out.print("World");
12.        }
13.    }
14. }
```

- a) Hello
- b) World
- c) HelloWorld
- d) Hello World

Answer: b

Explanation: System.out.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 caught by catch block printing just “World”.

Output:

```
$ javac exception_handling.java
$ java exception_handling
World
```

7. What will be the output of the following Java program?

```
1.    class exception_handling
2.    {
3.        public static void main(String args[])
4.        {
5.            try
6.            {
7.                int a, b;
8.                b = 0;
9.                a = 5 / b;
10.           System.out.print("A");
11.        }
12.        catch(ArithmeticException e)
13.        {
14.            System.out.print("B");
15.        }
16.    }
```

17. }

- a) A
- b) B
- c) Compilation Error
- d) Runtime Error

Answer: b

Explanation: None.

Output:

```
$ javac exception_handling.java
$ java exception_handling
B
```

8. What will be the output of the following Java program?

```
1.  class exception_handling
2.  {
3.      public static void main(String args[])
4.      {
5.          try
6.          {
7.              int a, b;
8.              b = 0;
9.              a = 5 / b;
10.             System.out.print("A");
11.         }
12.         catch(ArithmeticException e)
13.         {
14.             System.out.print("B");
15.         }
16.         finally
17.         {
18.             System.out.print("C");
19.         }
20.     }
21. }
```

- a) A
- b) B
- c) AC
- d) BC

Answer: d

Explanation: finally keyword is used to execute the code before try and catch block end.

Output:

```
$ javac exception_handling.java
$ java exception_handling
BC
```

9. What will be the output of the following Java program?

```
1.  class exception_handling
2.  {
3.      public static void main(String args[])
4.      {
5.          try
6.          {
7.              int i, sum;
8.              sum = 10;
9.              for (i = -1; i < 3 ;++i)
10.                 sum = (sum / i);
11.          }
12.          catch(ArithmeticException e)
13.          {
14.              System.out.print("0");
15.          }
16.          System.out.print(sum);
17.      }
18.  }
```

a) 0

b) 05

c) Compilation Error

d) Runtime Error

Answer: c

Explanation: Value of variable sum is printed outside of try block, sum is declared only in try block, outside try block it is undefined.

Output:

```
$ javac exception_handling.java
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    sum cannot be resolved to a variable
```

1. Which of the following keywords is used for throwing exception manually?

a) finally

b) try

c) throw

d) catch

Answer: c

Explanation: “throw” keyword is used for throwing exception manually in java program. User defined exceptions can be thrown too.

2. Which of the following classes can catch all exceptions which cannot be caught?

- a) RuntimeException**
- b) Error**
- c) Exception**
- d) ParentException**

Answer: b

Explanation: Runtime errors cannot be caught generally. Error class is used to catch such errors/exceptions.

3. Which of the following is a super class of all exception type classes?

- a) Catchable**
- b) RuntimeExceptions**
- c) String**
- d) Throwable**

Answer: d

Explanation: Throwable is built in class and all exception types are subclass of this class. It is the super class of all exceptions.

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

- a) thrown**
- b) alloc**
- c) malloc**
- d) new**

Answer: d

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

5. Which of the following keyword is used by calling function to handle exception thrown by called function?

- a) throws**
- b) throw**
- c) try**
- d) catch**

Answer: a

Explanation: A method specifies behaviour of being capable of causing exception. Throws clause in the method declaration guards caller of the method from exception.

6. Which of the following handles the exception when a catch is not used?

- a) finally**
- b) throw handler**
- c) default handler**
- d) java run time system**

Answer: c

Explanation: Default handler is used to handle all the exceptions if catch is not used to handle exception. Finally is called in any case.

7. Which part of code gets executed whether exception is caught or not?

- a) finally**
- b) try**

- c) catch
- d) throw

Answer: a

Explanation: Finally block of the code gets executed regardless exception is caught or not. File close, database connection close, etc are usually done in finally.

8. Which of the following should be true of the object thrown by a throw statement?

- a) Should be assignable to String type
- b) Should be assignable to Exception type
- c) Should be assignable to Throwable type
- d) Should be assignable to Error type

Answer: c

Explanation: The throw statement should be assignable to the throwable type. Throwable is the super class of all exceptions.

Answer: b

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

1. Which of these is a super class of all exceptional type classes?

- a) String
- b) RuntimeExceptions
- c) Throwable
- d) Cacheable

Answer: c

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

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

- a) Error
- b) Exception
- c) RuntimeException
- d) All of the mentioned

Answer: b

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

3. Which of these class is related to all the exceptions that cannot be caught?

- a) Error
- b) Exception
- c) RuntimeException
- d) All of the mentioned

Answer: a

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

4. Which of these handles the exception when no catch is used?

- a) Default handler
- b) finally
- c) throw handler
- d) Java run time system

Answer: a

Explanation: None.

5. What exception thrown by parseInt() method?

- a) ArithmeticException

- b) **ClassNotFoundException**
- c) **NullPointerException**
- d) **NumberFormatException**

Answer: d

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

6. What will be the output of the following Java code?

```
1.      class exception_handling
2.      {
3.          public static void main(String args[])
4.          {
5.              try
6.              {
7.                  System.out.print("Hello" + " " + 1 / 0);
8.              }
9.              finally
10.             {
11.                 System.out.print("World");
12.             }
13.         }
14.     }
```

- a) **Hello**
- b) **World**
- c) **Compilation Error**
- d) **First Exception then World**

Answer: d

Explanation: None.

Output:

```
$ javac exception_handling.java
$ java exception_handling
Exception in thread "main" java.lang.ArithmeticException: / by zero
World
```

7. What will be the output of the following Java code?

```
1.      class exception_handling
2.      {
3.          public static void main(String args[])
4.          {
5.              try
```

```

6.      {
7.          int i, sum;
8.          sum = 10;
9.          for (i = -1; i < 3 ;++i)
10.         {
11.             sum = (sum / i);
12.             System.out.print(i);
13.         }
14.     }
15.     catch(ArithmeticException e)
16.     {
17.         System.out.print("0");
18.     }
19. }
20. }

```

- a) -1
- b) 0
- c) -10
- d) -101

Answer: c

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

Output:

```

$ javac exception_handling.java
$ java exception_handling
-10

```

1. Which of these keywords is used to generate an exception explicitly?

- a) try
- b) finally
- c) throw
- d) catch

Answer: c

Explanation: None.

2. Which of these class is related to all the exceptions that are explicitly thrown?

- a) Error
- b) Exception
- c) Throwable
- d) Throw

Answer: c

Explanation: None.

3. Which of these operator is used to generate an instance of an exception than can be thrown by using throw?

- a) new

- b) malloc
- c) alloc
- d) thrown

Answer: a

Explanation: new is used to create an instance of an exception. All of java's built in run-time exceptions have two constructors: one with no parameters and one that takes a string parameter.

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

- a) try
- b) throw
- c) throws
- d) catch

Answer: c

Explanation: 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.

5. What will be the output of the following Java code?

```
1.      class exception_handling
2.      {
3.          public static void main(String args[])
4.          {
5.              try
6.              {
7.                  int a = args.length;
8.                  int b = 10 / a;
9.                  System.out.print(a);
10.             try
11.             {
12.                 if (a == 1)
13.                     a = a / a - a;
14.                 if (a == 2)
15.                 {
16.                     int []c = {1};
17.                     c[8] = 9;
18.                 }
19.             }
20.             catch (ArrayIndexOutOfBoundsException e)
21.             {
```



```

22.         System.out.println("TypeA");
23.     }
24.     catch (ArithmeticException e)
25.     {
26.         System.out.println("TypeB");
27.     }
28. }
29. }
30. }

```

- a) TypeA
- b) TypeB
- c) Compile Time Error
- d) 0TypeB

Answer: c

Explanation: Because we can't go beyond array limit

6. What will be the output of the following Java code?

```

1.  class exception_handling
2.  {
3.      public static void main(String args[])
4.      {
5.          try
6.          {
7.              System.out.print("A");
8.              throw new NullPointerException ("Hello");
9.          }
10.         catch(ArithmeticException e)
11.         {
12.             System.out.print("B");
13.         }
14.     }
15. }

```

- a) A
- b) B
- c) Hello
- d) Runtime Exception

Answer: d

Explanation: None.

Output:

```
$ javac exception_handling.java
$ java exception_handling
Exception in thread "main" java.lang.NullPointerException: Hello
    at exception_handling.main
```

7. What will be the output of the following Java code?

```
1. public class San
2. {
3.     public static void main(String[] args)
4.     {
5.         try
6.         {
7.             return;
8.         }
9.         finally
10.        {
11.            System.out.println( "Finally" );
12.        }
13.    }
14. }
```

- a) Finally**
- b) Compilation fails**
- c) The code runs with no output**
- d) An exception is thrown at runtime**

Answer: a

Explanation: Because finally will execute always.

8. What will be the output of the following Java code?

```
1. public class San
2. {
3.     public static void main(String args[])
4.     {
5.         try
6.         {
7.             System.out.print("Hello world ");
8.         }
9.         finally
```

```

10.      {
11.          System.out.println("Finally executing ");
12.      }
13.  }
14. }

```

- a) The program will not compile because no exceptions are specified
- b) The program will not compile because no catch clauses are specified
- c) Hello world
- d) Hello world Finally executing

Answer: d

Explanation: None

1. Which of these clause will be executed even if no exceptions are found?

- a) throws
- b) finally
- c) throw
- d) catch

Answer: b

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

2. A single try block must be followed by which of these?

- a) finally
- b) catch
- c) finally & catch
- d) none of the mentioned

Answer: c

Explanation: 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.

3. Which of these exceptions handles the divide by zero error?

- a) ArithmeticException
- b) MathException
- c) IllegalAccessException
- d) IllegarException

Answer: a

Explanation: None.

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

- a) ArithmeticException
- b) ArrayException
- c) ArrayIndexException
- d) ArrayIndexOutOfBoundsException

Answer: d

Explanation: 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.

5. What will be the output of the following Java code?

```
1.      class exception_handling
2.      {
3.          public static void main(String args[])
4.          {
5.              try
6.              {
7.                  int a = args.length;
8.                  int b = 10 / a;
9.                  System.out.print(a);
10.             }
11.             catch (ArithmeticException e)
12.             {
13.                 System.out.println("1");
14.             }
15.         }
16.     }
```

Note : Execution command line : \$ java exception_handling

a) 0

b) 1

c) Compilation Error

d) Runtime Error

Answer: b

Explanation: None.

Output:

```
$ javac exception_handling.java
$ java exception_handling
1
```

6. What will be the output of the following Java code?

```
1.      class exception_handling
2.      {
3.          public static void main(String args[])
4.          {
5.              try
6.              {
7.                  throw new NullPointerException ("Hello");
8.              }
9.              catch (ArithmeticException e)
```

```

10.         {
11.             System.out.print("B");
12.         }
13.     }
14. }

```

- a) A
- b) B
- c) Compilation Error
- d) Runtime Error

Answer: d

Explanation: 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.

Output:

```

$ javac exception_handling.java
$ java exception_handling
Exception in thread "main" java.lang.NullPointerException: Hello

```

7. What will be the output of the following Java code?

```

1. class exception_handling
2.     {
3.         public static void main(String args[])
4.         {
5.             try
6.             {
7.                 int a = 1;
8.                 int b = 10 / a;
9.                 try
10.                {
11.                    if (a == 1)
12.                        a = a / a - a;
13.                    if (a == 2)
14.                    {
15.                        int c[] = {1};
16.                        c[8] = 9;
17.                    }
18.                }
19.            finally
20.            {

```

```

21.             System.out.print("A");
22.         }
23.     }
24.     catch (Exception e)
25.     {
26.         System.out.println("B");
27.     }
28. }
29. }

```

- a) A
- b) B
- c) AB
- d) BA

Answer: a

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

Output:

```

$ javac exception_handling.java
$ java exception_handling
A

```

8. What will be the output of the following Java code?

```

1.     class exception_handling
2.     {
3.         public static void main(String args[])
4.         {
5.             try
6.             {
7.                 int a = args.length;
8.                 int b = 10 / a;
9.                 System.out.print(a);
10.            try
11.            {
12.                if (a == 1)
13.                    a = a / a - a;
14.                if (a == 2)
15.                {

```

```

16.                int []c = {1};
17.                c[8] = 9;
18.            }
19.        }
20.        catch (ArrayIndexOutOfBoundsException e)
21.        {
22.            System.out.println("TypeA");
23.        }
24.        catch (ArithmeticException e)
25.        {
26.            System.out.println("TypeB");
27.        }
28.    }
29. }

```

Note: Execution command line: \$ java exception_handling one two

- a) TypeA**
- b) TypeB**
- c) Compilation Error**
- d) Runtime Error**

Answer: c

Explanation: try without catch or finally

Output:

```

$ javac exception_handling.java
$ java exception_handling
Main.java:9: error: 'try' without 'catch', 'finally' or resource declarations

```

1. What is the use of try & catch?

- a) It allows us to manually handle the exception**
- b) It allows to fix errors**
- c) It prevents automatic terminating of the program in cases when an exception occurs**
- d) All of the mentioned**

Answer: d

Explanation: None.

2. Which of these keywords are used for the block to be examined for exceptions?

- a) try**
- b) catch**
- c) throw**
- d) check**

Answer: a

Explanation: try is used for the block that needs to be checked for exception.

3. Which of these keywords are used for the block to handle the exceptions generated by try block?

- a) try**

- b) catch
- c) throw
- d) check

Answer: b

Explanation: None.

4. Which of these keywords are used for generating an exception manually?

- a) try
- b) catch
- c) throw
- d) check

Answer: c

Explanation: None.

5. Which of these statements is incorrect?

- a) try block need not to be followed by catch block
- b) try block can be followed by finally block instead of catch block
- c) try can be followed by both catch and finally block
- d) try need not to be followed by anything

Answer: d

Explanation: try must be followed by either catch or finally block.

6. What will be the output of the following Java code?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              try
6.              {
7.                  int a = 0;
8.                  int b = 5;
9.                  int c = b / a;
10.                 System.out.print("Hello");
11.             }
12.             catch(Exception e)
13.             {
14.                 System.out.print("World");
15.             }
16.         }
17.     }
```

- a) Hello
- b) World

c) HelloWorld

d) Compilation Error

Answer: b

Explanation: None.

Output:

```
$ javac Output.javac
java Output
World
```

7. What will be the output of the following Java code?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.          {
5.              try
6.              {
7.                  int a = 0;
8.                  int b = 5;
9.                  int c = a / b;
10.                 System.out.print("Hello");
11.             }
12.             catch(Exception e)
13.             {
14.                 System.out.print("World");
15.             }
16.         }
17.     }
```

a) Hello

b) World

c) HelloWorld

d) Compilation Error

Answer: a

Explanation: None.

Output:

```
$ javac Output.javac
java Output
Hello
```

8. What will be the output of the following Java code?

```
1.      class Output
```

```
2.      {
3.          public static void main(String args[])
4.      {
5.          try
6.          {
7.              int a = 0;
8.              int b = 5;
9.              int c = b / a;
10.             System.out.print("Hello");
11.         }
12.     }
13. }
```

- a) Hello
- b) World
- c) HelloWorld
- d) Compilation Error

Answer: d

Explanation: try must be followed by either catch or finally

Output:

```
$ javac Output.javac
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Syntax error, insert "Finally" to complete BlockStatements
```

9. What will be the output of the following Java code?

```
1.      class Output
2.      {
3.          public static void main(String args[])
4.      {
5.          try
6.          {
7.              int a = 0;
8.              int b = 5;
9.              int c = a / b;
10.             System.out.print("Hello");
11.         }
12.         finally
13.     {
```

```
14.         System.out.print("World");
15.     }
16. }
17. }
```

- a) Hello
- b) World
- c) HelloWorld
- d) Compilation Error

Answer: c

Explanation: finally block is always executed after try block, no matter exception is found or not.

Output:

```
$ javac Output.javac
java Output
HelloWorld
```

10. What will be the output of the following Java code?

```
1.  class Output
2.  {
3.      public static void main(String args[])
4.      {
5.          try
6.          {
7.              int a = 0;
8.              int b = 5;
9.              int c = b / a;
10.         System.out.print("Hello");
11.     }
12.     catch(Exception e)
13.     {
14.         System.out.print("World");
15.     }
16.     finally
17.     {
18.         System.out.print("World");
19.     }
20. }
21. }
```

- a) Hello
- b) World
- c) HelloWorld
- d) WorldWorld

Answer: d

Explanation: finally block is always executed after try block, no matter exception is found or not. catch block is executed only when exception is found. Here divide by zero exception is found hence both catch and finally are executed.

Output:

```
$ javac Output.java
java Output
WorldWorld
```

1. Which of these classes is used to define exceptions?

- a) Exception
- b) Throwable
- c) Abstract
- d) System

Answer: a

Explanation: None.

2. Which of these methods return description of an exception?

- a) getException()
- b) getMessage()
- c) obtainDescription()
- d) obtainException()

Answer: b

Explanation: getMessage() returns a description of the exception.

3. Which of these methods is used to print stack trace?

- a) obtainStackTrace()
- b) printStackTrace()
- c) getStackTrace()
- d) displayStackTrace()

Answer: b

Explanation: None.

4. Which of these methods return localized description of an exception?

- a) getLocalizedMessage()
- b) getMessage()
- c) obtainLocalizedMessage()
- d) printLocalizedMessage()

Answer: a

Explanation: None.

5. Which of these classes is super class of Exception class?

- a) Throwable
- b) System
- c) RunTime
- d) Class

Answer: a

Explanation: None.

6. What will be the output of the following Java code?

```
1.    class Myexception extends Exception
2.    {
3.        int detail;
4.        Myexception(int a)
5.        {
6.            detail = a;
7.        }
8.        public String toString()
9.        {
10.            return "detail";
11.        }
12.    }
13.    class Output
14.    {
15.        static void compute (int a) throws Myexception
16.        {
17.            throw new Myexception(a);
18.        }
19.        public static void main(String args[])
20.        {
21.            try
22.            {
23.                compute(3);
24.            }
25.            catch(Myexception e)
26.            {
27.                System.out.print("Exception");
28.            }
29.        }
30.    }
```

- a) 3
- b) Exception
- c) Runtime Error
- d) Compilation Error

Answer: b

Explanation: Myexception is self defined exception.

Output:

```
$ javac Output.java
java Output
Exception
```

7. What will be the output of the following Java code?

```
1.      class Myexception extends Exception
2.      {
3.          int detail;
4.          Myexception(int a)
5.          {
6.              detail = a;
7.          }
8.          public String toString()
9.          {
10.             return "detail";
11.          }
12.      }
13.      class Output
14.      {
15.          static void compute (int a) throws Myexception
16.          {
17.              throw new Myexception(a);
18.          }
19.          public static void main(String args[])
20.          {
21.              try
22.              {
23.                  compute(3);
24.              }
25.              catch (DevideByZeroException e)
26.              {
27.                  System.out.print("Exception");
28.              }
```

```
29.         }
```

```
30.     }
```

- a) 3
- b) Exception
- c) Runtime Error
- d) Compilation Error

Answer: c

Explanation: Mexception is self defined exception, we are generating Myexception but catching DevideByZeroException which causes error.

Output:

```
$ javac Output.javac
```

8. What will be the output of the following Java code?

```
1.     class exception_handling
2.     {
3.         public static void main(String args[])
4.         {
5.             try
6.             {
7.                 throw new NullPointerException ("Hello");
8.                 System.out.print("A");
9.             }
10.            catch(ArithmeticException e)
11.            {
12.                System.out.print("B");
13.            }
14.        }
15.    }
```

- a) A
- b) B
- c) Compilation Error
- d) Runtime Error

Answer: d

Explanation: 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.

Output:

```
$ javac exception_handling.java
$ java exception_handling
Exception in thread "main" java.lang.NullPointerException: Hello
```

9. What will be the output of the following Java code?

```
1.      class Myexception extends Exception
2.      {
3.          int detail;
4.          Myexception(int a)
5.          {
6.              detail = a;
7.          }
8.          public String toString()
9.          {
10.              return "detail";
11.          }
12.      }
13.      class Output
14.      {
15.          static void compute (int a) throws Myexception
16.          {
17.              throw new Myexception(a);
18.          }
19.          public static void main(String args[])
20.          {
21.              try
22.              {
23.                  compute(3);
24.              }
25.              catch(Exception e)
26.              {
27.                  System.out.print("Exception");
28.              }
29.          }
30.      }
```

a) 3

b) Exception

c) Runtime Error

d) Compilation Error

Answer: b

Explanation: Myexception is self defined exception.

Output:

```
$ javac Output.java
java Output
Exception
```

10. What will be the output of the following Java code?

```
1.  class exception_handling
2.  {
3.      public static void main(String args[])
4.      {
5.          try
6.          {
7.              int a = args.length;
8.              int b = 10 / a;
9.              System.out.print(a);
10.         try
11.         {
12.             if (a == 1)
13.                 a = a / a - a;
14.             if (a == 2)
15.             {
16.                 int c = {1};
17.                 c[8] = 9;
18.             }
19.         }
20.         catch (ArrayIndexOutOfBoundsException e)
21.         {
22.             System.out.println("TypeA");
23.         }
24.         catch (ArithmeticException e)
25.         {
26.             System.out.println("TypeB");
27.         }
28.     }
29. }
```

Note : Execution command line : \$ java exception_handling one

- a) TypeA
- b) TypeB
- c) Compilation Error
- d) Runtime Error

Answer: c

Explanation: try without catch or finally

Output:

```
$ javac exception_handling.java
$ java exception_handling
error: 'try' without 'catch', 'finally' or resource declarations
```

1. Which of this method can be used to make the main thread to be executed last among all the threads?

- a) stop()
- b) sleep()
- c) join()
- d) call()

Answer: b

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

2. Which of this method is used to find out that a thread is still running or not?

- a) run()
- b) Alive()
- c) isAlive()
- d) checkRun()

Answer: c

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

3. What is the default value of priority variable MIN_PRIORITY AND MAX_PRIORITY?

- a) 0 & 256
- b) 0 & 1
- c) 1 & 10
- d) 1 & 256

Answer: c

Explanation: None.

4. Which of these method waits for the thread to terminate?

- a) sleep()
- b) isAlive()
- c) join()
- d) stop()

Answer: c

Explanation: None.

5. Which of these method is used to explicitly set the priority of a thread?

- a) set()
- b) make()
- c) setPriority()

d) makePriority()

Answer: c

Explanation: 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().

6. What is synchronization in reference to a thread?

a) It's a process of handling situations when two or more threads need access to a shared resource

b) It's a process by which many thread are able to access same shared resource simultaneously

c) It's a process by which a method is able to access many different threads simultaneously

d) It's a method that allow too many threads to access any information require

Answer: a

Explanation: 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

7. What will be the output of the following Java code?

```
1.      class newthread extends Thread
2.      {
3.          newthread()
4.          {
5.              super("My Thread");
6.              start();
7.          }
8.          public void run()
9.          {
10.             System.out.println(this);
11.         }
12.     }
13.     class multithreaded_programing
14.     {
15.         public static void main(String args[])
16.         {
17.             new newthread();
18.         }
19.     }
```

a) My Thread

b) Thread[My Thread,5,main]

c) Compilation Error

d) Runtime Error

Answer: b

Explanation: Although we have not created any object of thread class still we can make a thread pointing to main

method, we can refer it by using this.

Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
Thread[My Thread,5,main].
```

8. What will be the output of the following Java code?

```
1.      class newthread extends Thread
2.      {
3.          Thread t;
4.          newthread()
5.          {
6.              t = new Thread(this,"My Thread");
7.              t.start();
8.          }
9.          public void run()
10.         {
11.             try
12.             {
13.                 t.join()
14.                 System.out.println(t.getName());
15.             }
16.             catch(Exception e)
17.             {
18.                 System.out.print("Exception");
19.             }
20.         }
21.     }
22.     class multithreaded_programing
23.     {
24.         public static void main(String args[])
25.         {
26.             new newthread();
27.         }
28.     }
```

a) My Thread

b) Thread[My Thread,5,main]

c) Exception

d) Runtime Error

Answer: d

Explanation: join() method of Thread class waits for thread being called to finish or terminate, but here we have no condition which can terminate the thread, hence code 't.join()' leads to runtime error and nothing will be printed on the screen.

Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
```

9. What will be the output of the following Java code?

```
1.      class newthread extends Thread
2.      {
3.          Thread t;
4.          newthread()
5.          {
6.              t= new Thread(this,"New Thread");
7.              t.start();
8.          }
9.          public void run()
10.         {
11.             System.out.println(t.isAlive());
12.         }
13.     }
14.     class multithreaded_programing
15.     {
16.         public static void main(String args[])
17.         {
18.             new newthread();
19.         }
20.     }
```

a) 0

b) 1

c) true

d) false

Answer: c

Explanation: isAlive() method is used to check whether the thread being called is running or not, here thread is the main() method which is running till the program is terminated hence it returns true.

Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
true
```

10. What will be the output of the following Java code?

```
1.      class newthread extends Thread
2.      {
3.          Thread t1,t2;
4.          newthread()
5.          {
6.              t1 = new Thread(this,"Thread_1");
7.              t2 = new Thread(this,"Thread_2");
8.              t1.start();
9.              t2.start();
10.         }
11.         public void run()
12.         {
13.             t2.setPriority(Thread.MAX_PRIORITY);
14.             System.out.print(t1.equals(t2));
15.         }
16.     }
17.     class multithreaded_programing
18.     {
19.         public static void main(String args[])
20.         {
21.             new newthread();
22.         }
23.     }
```

- a) true
- b) false
- c) true true
- d) false false

Answer: d

Explanation: This program was previously done by using Runnable interface, here we have used Thread class. This shows both the methods are equivalent, we can use any of them to create a thread.

Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
falsefalse
```

1. Which of these method is used to implement Runnable interface?

- a) stop()
- b) run()
- c) runThread()
- d) stopThread()

Answer: b

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

2. Which of these method is used to begin the execution of a thread?

- a) run()
- b) start()
- c) runThread()
- d) startThread()

Answer: b

Explanation: None.

3. Which of these statement is incorrect?

- a) A thread can be formed by implementing Runnable interface only
- b) A thread can be formed by a class that extends Thread class
- c) start() method is used to begin execution of the thread
- d) run() method is used to begin execution of a thread before start() method in special cases

Answer: d

Explanation: 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.

4. What will be the output of the following Java code?

```
1.      class newthread implements Runnable
2.      {
3.          Thread t;
4.          newthread()
5.          {
6.              t = new Thread(this, "My Thread");
7.              t.start();
8.          }
9.          public void run()
10.         {
11.             System.out.println(t.getName());
12.         }
13.     }
14.     class multithreaded_programing
15.     {
16.         public static void main(String args[])
```

```
17.         {  
18.             new newthread();  
19.         }  
20.     }
```

- a) My Thread
- b) Thread[My Thread,5,main]
- c) Compilation Error
- d) Runtime Error

Answer: a

Explanation: None.

Output:

```
$ javac multithreaded_programing.java  
$ java multithreaded_programing  
My Thread
```

5. What will be the output of the following Java code?

```
1.     class newthread implements Runnable  
2.     {  
3.         Thread t;  
4.         newthread()  
5.         {  
6.             t = new Thread(this,"My Thread");  
7.             t.start();  
8.         }  
9.         public void run()  
10.        {  
11.            System.out.println(t);  
12.        }  
13.    }  
14.    class multithreaded_programing  
15.    {  
16.        public static void main(String args[])  
17.        {  
18.            new newthread();  
19.        }  
20.    }
```

- a) My Thread
- b) Thread[My Thread,5,main]

c) Compilation Error
d) Runtime Error

Answer: b
Explanation: None.
Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
Thread[My Thread,5,main]
```

6. What will be the output of the following Java code?

```
1.      class newthread implements Runnable
2.      {
3.          Thread t;
4.          newthread()
5.          {
6.              t = new Thread(this,"My Thread");
7.              t.start();
8.          }
9.      }
10.     class multithreaded_programing
11.     {
12.         public static void main(String args[])
13.         {
14.             new newthread();
15.         }
16.     }
```

- a) My Thread**
- b) Thread[My Thread,5,main]**
- c) Compilation Error**
- d) Runtime Error**

Answer: c
Explanation: Thread t has been made by using Runnable interface, hence it is necessary to use inherited abstract method run() method to specify instructions to be implemented on the thread, since no run() method is used it gives a compilation error.
Output:

```
$ javac multithreaded_programing.java
The type newthread must implement the inherited abstract method Runnable.run()
```

7. What will be the output of the following Java code?

```
1.      class newthread implements Runnable
2.      {
```

```

3.         Thread t;
4.
5.         newthread()
6.         {
7.             t = new Thread(this,"New Thread");
8.             t.start();
9.         }
10.        public void run()
11.        {
12.            t.setPriority(Thread.MAX_PRIORITY);
13.            System.out.println(t);
14.        }
15.        class multithreaded_programing
16.        {
17.            public static void main(String args[])
18.            {
19.                new newthread();
20.            }
21.        }

```

- a) Thread[New Thread,0,main]
- b) Thread[New Thread,1,main]
- c) Thread[New Thread,5,main]
- d) Thread[New Thread,10,main]

Answer: d

Explanation: Thread t has been made with default priority value 5 but in run method the priority has been explicitly changed to MAX_PRIORITY of class thread, that is 10 by code 't.setPriority(Thread.MAX_PRIORITY);' using the setPriority function of thread t.

Output:

```

$ javac multithreaded_programing.java
$ java multithreaded_programing
Thread[New Thread,10,main]

```

8. What will be the output of the following Java code?

```

1.        class newthread implements Runnable
2.        {
3.            Thread t;
4.            newthread()
5.            {

```

```

6.         t1 = new Thread(this, "Thread_1");
7.
8.         t2 = new Thread(this, "Thread_2");
9.
10.        t1.start();
11.
12.        t2.start();
13.
14.    }
15.
16.    public void run()
17.    {
18.
19.        t2.setPriority(Thread.MAX_PRIORITY);
20.
21.        System.out.print(t1.equals(t2));
22.
23.    }
24.
25.    class multithreaded_programing
26.    {
27.
28.        public static void main(String args[])
29.        {
30.
31.            new newthread();
32.
33.        }
34.
35.    }

```

- a) true
- b) false
- c) true true
- d) false false

Answer: d

Explanation: Threads t1 & t2 are created by class newthread that is implementing runnable interface, hence both the threads are provided their own run() method specifying the actions to be taken. When constructor of newthread class is called first the run() method of t1 executes than the run method of t2 printing 2 times “false” as both the threads are not equal one is having different priority than other, hence falsefalse is printed.

Output:

```

$ javac multithreaded_programing.java
$ java multithreaded_programing
falsefalse

```

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

- a) get()
- b) ThreadPriority()
- c) getPriority()
- d) getThreadPriority()

Answer: c

Explanation: None.

2. Which of these method of Thread class is used to Suspend a thread for a period of time?

- a) sleep()

- b) terminate()
- c) suspend()
- d) stop()

Answer: a

Explanation: None.

3. Which function of pre defined class Thread is used to check weather current thread being checked is still running?

- a) isAlive()
- b) Join()
- c) isRunning()
- d) Alive()

Answer: a

Explanation: isAlive() function is defined in class Thread, it is used for implementing multithreading and to check whether the thread called upon is still running or not.

4. What will be the output of the following Java code?

```
1.      class multithreaded_programing
2.      {
3.
4.          public static void main(String args[])
5.          {
6.
7.              Thread t = Thread.currentThread();
8.
9.              t.setName("New Thread");
10.
11.              System.out.println(t);
12.
13.          }
14.      }
```

- a) Thread[5,main]
- b) Thread[New Thread,5]
- c) Thread[main,5,main]
- d) Thread[New Thread,5,main]

Answer: d

Explanation: None.

Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
Thread[New Thread,5,main]
```

5. What is the priority of the thread in output in the following Java program?

```
1.      class multithreaded_programing
2.      {
3.
4.          public static void main(String args[])
5.          {
6.
7.              Thread t = Thread.currentThread();
8.
9.              t.setName("New Thread");
10.
11.          }
12.      }
```

```
7.         System.out.println(t.getName());
8.     }
9. }
```

- a) main
- b) Thread
- c) New Thread
- d) Thread[New Thread,5,main]

Answer: c

Explanation: The getName() function is used to obtain the name of the thread, in this code the name given to thread is 'New Thread'.

Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
New Thread
```

6. What is the name of the thread in output in the following Java program?

```
1.     class multithreaded_programing
2.     {
3.         public static void main(String args[])
4.         {
5.             Thread t = Thread.currentThread();
6.             System.out.println(t.getPriority());
7.         }
8.     }
```

- a) 0
- b) 1
- c) 4
- d) 5

Answer: d

Explanation: The default priority given to a thread is 5.

Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
5
```

7. What is the name of the thread in output in the following Java program?

```
1.     class multithreaded_programing
2.     {
3.         public static void main(String args[])
4.         {
5.             Thread t = Thread.currentThread();
```

```
6.         System.out.println(t.isAlive());
7.     }
8. }
```

- a) 0
- b) 1
- c) true
- d) false

Answer: c

Explanation: Thread t is seeded to currently program, hence when you run the program the thread becomes active & code 't.isAlive' returns true.

Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
true
```

1. What is multithreaded programming?

- a) It's a process in which two different processes run simultaneously
- b) It's a process in which two or more parts of same process run simultaneously
- c) It's a process in which many different process are able to access same information
- d) It's a process in which a single process can access information from many sources

Answer: b

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

2. Which of these are types of multitasking?

- a) Process based
- b) Thread based
- c) Process and Thread based
- d) None of the mentioned

Answer: c

Explanation: There are two types of multitasking: Process based multitasking and Thread based multitasking.

3. Thread priority in Java is?

- a) Integer
- b) Float
- c) double
- d) long

Answer: a

Explanation: Java assigns to each thread a priority that determines hoe that thread should be treated with respect to others. Thread priority is integers that specify relative priority of one thread to another.

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

- a) Anyone will be executed first lexographically
- b) Both of them will be executed simultaneously
- c) None of them will be executed
- d) It is dependent on the operating system

Answer: d

Explanation: 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.

5. Which of these statements is incorrect?

- a) By multithreading CPU idle time is minimized, and we can take maximum use of it
- b) By multitasking CPU idle time is minimized, and we can take maximum use of it
- c) Two thread in Java can have the same priority
- d) A thread can exist only in two states, running and blocked

Answer: d

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

6. What will be the output of the following Java code?

```
1.      class multithreaded_programing
2.      {
3.          public static void main(String args[])
4.          {
5.              Thread t = Thread.currentThread();
6.              System.out.println(t);
7.          }
8.      }
```

- a) Thread[5,main]
- b) Thread[main,5]
- c) Thread[main,0]
- d) Thread[main,5,main]

Answer: d

Explanation: None.

Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
Thread[main,5,main]
```

7. What is the priority of the thread in the following Java Program?

```
1.      class multithreaded_programing
2.      {
3.          public static void main(String args[])
4.          {
5.              Thread t = Thread.currentThread();
6.              System.out.println(t);
7.          }
8.      }
```

- a) 4
- b) 5
- c) 0
- d) 1

Answer: b

Explanation: The output of program is Thread[main,5,main], in this priority assigned to the thread is 5. It's the default value. Since we have not named the thread they are named by the group to they belong i:e main method.
Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
Thread[main,5,main]
```

8. What is the name of the thread in the following Java Program?

```
1.      class multithreaded_programing
2.      {
3.          public static void main(String args[])
4.          {
5.              Thread t = Thread.currentThread();
6.              System.out.println(t);
7.          }
8.      }
```

- a) main**
- b) Thread**
- c) System**
- d) None of the mentioned**

Answer: a
Explanation: The output of program is Thread[main,5,main], Since we have not explicitly named the thread they are named by the group to they belong i:e main method. Hence they are named 'main'.
Output:

```
$ javac multithreaded_programing.java
$ java multithreaded_programing
Thread[main,5,main]
```

1. What requires less resources?

- a) Thread**
- b) Process**
- c) Thread and Process**
- d) Neither Thread nor Process**

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

2. What does not prevent JVM from terminating?

- a) Process**
- b) Daemon Thread**
- c) User Thread**
- d) JVM Thread**

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

3. What decides thread priority?

- a) Process
- b) Process scheduler
- c) Thread
- d) Thread scheduler

Answer: d

Explanation: 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.

4. What is true about time slicing?

- a) Time slicing is OS service that allocates CPU time to available runnable thread
- b) Time slicing is the process to divide the available CPU time to available runnable thread
- c) Time slicing depends on its implementation in OS
- d) Time slicing allocates more resources to thread

Answer: b

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

Answer: a
Explanation: Deadlock is java programming situation where one thread waits for an object lock that is acquired by other thread and vice-versa.

6. What should not be done to avoid deadlock?

- a) Avoid using multiple threads
- b) Avoid hold several locks at once
- c) Execute foreign code while holding a lock
- d) Use interruptible locks

Answer: c

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

7. What is true about threading?

- a) run() method calls start() method and runs the code
- b) run() method creates new thread
- c) run() method can be called directly without start() method being called
- d) start() method creates new thread and calls code written in run() method

Answer: d

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

8. Which of the following is a correct constructor for thread?

- a) Thread(Runnable a, String str)
- b) Thread(int priority)
- c) Thread(Runnable a, int priority)
- d) Thread(Runnable a, ThreadGroup t)

Answer: a

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

9. Which of the following stops execution of a thread?

- a) Calling SetPriority() method on a Thread object
- b) Calling notify() method on an object
- c) Calling wait() method on an object
- d) Calling read() method on an InputStream object

Answer: b

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

10. Which of the following will ensure the thread will be in running state?

- a) yield()
- b) notify()
- c) wait()
- d) Thread.killThread()

Answer: c

Explanation: 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.

1. Which of these keywords are used to implement synchronization?

- a) synchronize
- b) syn
- c) synch
- d) synchronized

Answer: d

Explanation: None.

2. Which of this method is used to avoid polling in Java?

- a) wait()
- b) notify()
- c) notifyAll()
- d) all of the mentioned

Answer: d

Explanation: 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().

3. Which 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?

- a) wait()
- b) notify()
- c) notifyAll()
- d) sleep()

Answer: a

Explanation: 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.

4. Which of these method wakes up the first thread that called wait()?

- a) wake()
- b) notify()
- c) start()
- d) notifyAll()

Answer: b

Explanation: None.

5. Which of these method wakes up all the threads?

- a) wakeAll()
- b) notify()
- c) start()
- d) notifyAll()

Answer: d

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

6. What is synchronization in reference to a thread?

- a) It's a process of handling situations when two or more threads need access to a shared resource
- b) It's a process by which many thread are able to access same shared resource simultaneously
- c) It's a process by which a method is able to access many different threads simultaneously
- d) It's a method that allow too many threads to access any information the require

Answer: a

Explanation: 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

7. What will be the output of the following Java program?

```
1.      class newthread extends Thread
2.      {
3.          Thread t;
4.          String name;
5.          newthread(String threadname)
6.          {
7.              name = threadname;
8.              t = new Thread(this,name);
9.              t.start();
10.         }
11.         public void run()
12.         {
13.         }
14.
15.     }
16.     class multithreaded_programing
17.     {
18.         public static void main(String args[])
19.         {
20.             newthread obj1 =      new newthread("one");
21.             newthread obj2 =      new newthread("two");
22.             try
23.             {
24.                 obj1.t.wait();
25.                 System.out.print(obj1.t.isAlive());
26.             }
27.             catch(Exception e)
```

```

28.         {
29.             System.out.print("Main thread interrupted");
30.         }
31.     }
32. }

```

- a) true
- b) false
- c) Main thread interrupted
- d) None of the mentioned

Answer: c

Explanation: `obj1.t.wait()` causes main thread to go out of processing in sleep state hence causes exception and “Main thread interrupted” is printed.

Output:

```

$ javac multithreaded_programing.java
$ java multithreaded_programing
Main thread interrupted

```

8. What will be the output of the following Java program?

```

1.     class newthread extends Thread
2.     {
3.         Thread t;
4.         String name;
5.         newthread(String threadname)
6.         {
7.             name = threadname;
8.             t = new Thread(this,name);
9.             t.start();
10.        }
11.        public void run()
12.        {
13.        }
14.
15.    }
16.    class multithreaded_programing
17.    {
18.        public static void main(String args[])
19.        {
20.            newthread obj1 =      new newthread("one");

```

```

21.         newthread obj2 =         new newthread("two");
22.
23.         try
24.             Thread.sleep(1000);
25.             System.out.print(obj1.t.isAlive());
26.         }
27.         catch(InterruptedException e)
28.         {
29.             System.out.print("Main thread interrupted");
30.         }
31.     }
32. }

```

- a) true
- b) false
- c) Main thread interrupted
- d) None of the mentioned

Answer: b

Explanation: Thread.sleep(1000) has caused all the threads to be suspended for some time, hence obj1.t.isAlive() returns false.

Output:

```

$ javac multithreaded_programing.java
$ java multithreaded_programing
false

```

9. What will be the output of the following Java program?

```

1.     class newthread extends Thread
2.     {
3.         Thread t;
4.         String name;
5.         newthread(String threadname)
6.         {
7.             name = threadname;
8.             t = new Thread(this,name);
9.             t.start();
10.        }
11.        public void run()
12.        {
13.        }

```

```

14.
15.     }
16.     class multithreaded_programing
17.     {
18.         public static void main(String args[])
19.         {
20.             newthread obj1 =      new newthread("one");
21.             newthread obj2 =      new newthread("two");
22.             try
23.             {
24.                 System.out.print(obj1.t.equals(obj2.t));
25.             }
26.             catch(Exception e)
27.             {
28.                 System.out.print("Main thread interrupted");
29.             }
30.         }
31.     }

```

- a) true
- b) false
- c) Main thread interrupted
- d) None of the mentioned

Answer: b

Explanation: Both obj1 and obj2 have threads with different name that is “one” and “two” hence obj1.t.equals(obj2.t) returns false.

Output:

```

$ javac multithreaded_programing.java
$ java multithreaded_programing
false

```

10. What will be the output of the following Java program?

```

1.     class newthread extends Thread
2.     {
3.         Thread t;
4.         newthread()
5.         {
6.             t1 = new Thread(this, "Thread_1");
7.             t2 = new Thread(this, "Thread_2");

```

```

8.         t1.start();
9.         t2.start();
10.    }
11.    public void run()
12.    {
13.        t2.setPriority(Thread.MAX_PRIORITY);
14.        System.out.print(t1.equals(t2));
15.    }
16. }
17. class multithreaded_programing
18. {
19.     public static void main(String args[])
20.     {
21.         new newthread();
22.     }
23. }

```

- a) true
- b) false
- c) true true
- d) false false

Answer: d

Explanation: This program was previously done by using Runnable interface, here we have used Thread class. This shows both the method are equivalent, we can use any of them to create a thread.

Output:

```

$ javac multithreaded_programing.java
$ java multithreaded_programing
falsefalse

```

1. What does AWT stands for?

- a) All Window Tools
- b) All Writing Tools
- c) Abstract Window Toolkit
- d) Abstract Writing Toolkit

Answer: c

Explanation: AWT stands for Abstract Window Toolkit, it is used by applets to interact with the user.

2. Which of these is used to perform all input & output operations in Java?

- a) streams
- b) Variables
- c) classes
- d) Methods

Answer: a

Explanation: Like in any other language, streams are used for input and output operations.

3. Which of these is a type of stream in Java?

- a) Integer stream
- b) Short stream
- c) Byte stream
- d) Long stream

Answer: c

Explanation: Java defines only two types of streams – Byte stream and character stream.

4. Which of these classes are used by Byte streams for input and output operation?

- a) InputStream
- b) OutputStream
- c) Reader
- d) All of the mentioned

Answer: a

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

5. Which of these classes are used by character streams for input and output operations?

- a) InputStream
- b) Writer
- c) ReadStream
- d) OutputStream

Answer: b

Explanation: Character streams uses Writer and Reader classes for input & output operations.

6. Which of these class is used to read from byte array?

- a) InputStream
- b) BufferedInputStream
- c) ByteArrayInputStream
- d) ByteArrayOutputStream

Answer: d

Explanation: None.

7. What will be the output of the following Java program if input given is 'abcqfghqbcd'?

```
1.    class Input_Output
2.    {
3.        public static void main(String args[]) throws IOException
4.        {
5.            char c;
6.            BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));
7.            do
8.            {
9.                c = (char) obj.read();
10.           System.out.print(c);
11.           } while(c != 'q');
12.    }
```



```
13.      }
```

- a) abcqfgh
- b) abc
- c) abcq
- d) abcqfghq

Answer: c

Explanation: None.

Output:

```
$ javac Input_Output.java
$ java Input_Output
abcq
```

8. What will be the output of the following Java program if input given is “abc’def’egh”?

```
1.      class Input_Output
2.      {
3.          public static void main(String args[]) throws IOException
4.          {
5.              char c;
6.              BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));
7.              do
8.              {
9.                  c = (char) obj.read();
10.                 System.out.print(c);
11.             } while(c!='\');
12.         }
13.     }
```

- a) abc’
- b) abcdef’
- c) abc’def’egh
- d) abcqfghq

Answer: a

Explanation: \’ is used for single quotes that is for representing ‘.

Output:

```
$ javac Input_Output.java
$ java Input_Output
abc’
```

9. What will be the output of the following Java program?

```
1.      class output
2.      {
3.          public static void main(String args[])
```

```

4.      {
5.          StringBuffer c = new StringBuffer("Hello");
6.          System.out.println(c.length());
7.      }
8.  }

```

- a) 4
- b) 5
- c) 6
- d) 7

Answer: b

Explanation: length() method is used to obtain length of StringBuffer object, length of "Hello" is 5.

Output:

```

$ javac output.java
$ java output
5

```

1. Which exception is thrown by read() method?

- a) IOException
- b) InterruptedException
- c) SystemException
- d) SystemInputException

Answer: a

Explanation: read method throws IOException.

2. Which of these is used to read a string from the input stream?

- a) get()
- b) getLine()
- c) read()
- d) readLine()

Answer: c

Explanation: None.

3. Which of these class is used to read characters and strings in Java from console?

- a) BufferedReader
- b) StringReader
- c) BufferedStreamReader
- d) InputStreamReader

Answer: a

Explanation: None.

4. Which of these class is implemented by FilterInputStream class?

- a) InputStream
- b) InputOutputStream
- c) BufferedInputStream
- d) SequenceInputStream

Answer: a

Explanation: FileInputStream implements InputStream.

5. What will be the output of the following Java program if input given is "Hello stop World"?

```

1.      class Input_Output
2.      {
3.          public static void main(String args[]) throws IOException
4.          {
5.              string str;
6.              BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));
7.              do
8.              {
9.                  str = (char) obj.readLine();
10.                 System.out.print(str);
11.             } while(!str.equals("strong"));
12.         }
13.     }

```

- a) Hello
- b) Hello stop
- c) World
- d) Hello stop World

Answer: d

Explanation: “stop” will be able to terminate the do-while loop only when it occurs singly in a line. “Hello stop World” does not terminate the loop.

Output:

```

$ javac Input_Output.java
$ java Input_Output
Hello stop World

```

6. What will be the output of the following Java program?

```

1.      class output
2.      {
3.          public static void main(String args[])
4.          {
5.              StringBuffer c = new StringBuffer("Hello");
6.              StringBuffer c1 = new StringBuffer(" World");
7.              c.append(c1);
8.              System.out.println(c);
9.          }
10.     }

```

- a) Hello
- b) World
- c) Helloworld

d) Hello World

Answer: d

Explanation: append() method of class StringBuffer is used to concatenate the string representation to the end of invoking string.

Output:

```
$ javac output.java
$ java output
Hello World
```

7. What will be the output of the following Java program?

```
1.      class output
2.      {
3.          public static void main(String args[])
4.          {
5.              StringBuffer s1 = new StringBuffer("Hello");
6.              s1.setCharAt(1,x);
7.              System.out.println(s1);
8.          }
9.      }
```

- a) xello
- b) xxxxx
- c) Hxlllo
- d) Hexlo

Answer: c

Explanation: None.

Output:

```
$ javac output.java
$ java output
Hxlllo
```

8. What will be the output of the following Java program if input given is “abc’def/’egh”?

```
1.      class Input_Output
2.      {
3.          public static void main(String args[]) throws IOException
4.          {
5.              char c;
6.              BufferedReader obj = new BufferedReader(new InputStreamReader(System.in));
7.              do
8.              {
9.                  c = (char) obj.read();
```

```

10.                System.out.print(c);

11.                } while(c != '\\');

12.            }

13.        }

```

- a) abc'
- b) abcdef'
- c) abc'def'egh
- d) abcqfghq

Answer: a

Explanation: \' is used for single quotes that is for representing ' .

Output:

```

$ javac Input_Output.java
$ java Input_Output
abc '

```

1. Which of these class contains the methods print() & println()?

- a) System
- b) System.out
- c) BUfferedOutputStream
- d) PrintStream

Answer: d

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

2. Which of these methods can be used to writing console output?

- a) print()
- b) println()
- c) write()
- d) all of the mentioned

Answer: d

Explanation: None.

3. Which of these classes are used by character streams output operations?

- a) InputStream
- b) Writer
- c) ReadStream
- d) InputOutputStream

Answer: b

Explanation: Character streams uses Writer and Reader classes for input & output operations.

4. Which of these class is used to read from a file?

- a) InputStream
- b) BufferedInputStream
- c) FileInputStream
- d) BufferedFileInputStream

Answer: c

Explanation: None.

5. What will be the output of the following Java program?

```

1.     class output
2.     {
3.         public static void main(String args[])
4.         {
5.             String a="hello i love java";
6.             System.out.println(indexof('i')+" "+indexOf('o')+" "+lastIndexof('i')+" "+lastIndexof('o'));
7.         }
8.     }

```

a) 6 4 6 9

b) 5 4 5 9

c) 7 8 8 9

d) 4 3 6 9

Answer: a

Explanation: indexOf('c') and lastIndexof('c') are pre defined function which are used to get the index of first and last occurrence of the character pointed by c in the given array.

Output:

```

$ javac output.java
$ java output
6 4 6 9

```

6. What will be the output of the following Java program?

```

1.     class output
2.     {
3.         public static void main(String args[])
4.         {
5.             char c[]={'a','1','b',' ','A','0'};
6.             for (int i = 0; i < 5; ++i)
7.             {
8.                 if(Character.isDigit(c[i]))
9.                     System.out.println(c[i] + " is a digit");
10.                if(Character.isWhitespace(c[i]))
11.                    System.out.println(c[i] + " is a Whitespace character");
12.                if(Character.isUpperCase(c[i]))
13.                    System.out.println(c[i] + " is an Upper case Letter");
14.                if(Character.isLowerCase(c[i]))
15.                    System.out.println(c[i] + " is a lower case Letter");
16.                i = i + 3;

```

```
17.         }
18.     }
19. }
```

7. What will be the output of the following Java program?

```
1.     class output
2.     {
3.         public static void main(String args[])
4.         {
5.             StringBuffer s1 = new StringBuffer("Hello");
6.             StringBuffer s2 = s1.reverse();
7.             System.out.println(s2);
8.         }
9.     }
```

- a) Hello
- b) olleH
- c) HelloolleH
- d) olleHHello

Answer: b
Explanation: reverse() method reverses all characters. It returns the reversed object on which it was called.
Output:

```
$ javac output.java
$ java output
olleH
```

1. Which of these class contains the methods used to write in a file?

- a) FileStream
- b) FileInputStream
- c) BUfferedOutputStream
- d) FileBufferStream

Answer: b
Explanation: None.

2. Which of these exception is thrown in cases when the file specified for writing is not found?

- a) IOException
- b) FileException
- c) FileNotFoundException
- d) FileInputException

Answer: c
Explanation: In cases when the file specified is not found, then FileNotFoundException is thrown by java run-time system, earlier versions of java used to throw IOException but after Java 2.0 they throw FileNotFoundException.

3. Which of these methods are used to read in from file?

- a) get()
- b) read()
- c) scan()

d) readFileInput()

Answer: b

Explanation: Each time read() is called, it reads a single byte from the file and returns the byte as an integer value. read() returns -1 when the end of the file is encountered.

4. Which of these values is returned by read() method is end of file (EOF) is encountered?

- a) 0**
- b) 1**
- c) -1**
- d) Null**

Answer: c

Explanation: Each time read() is called, it reads a single byte from the file and returns the byte as an integer value. read() returns -1 when the end of the file is encountered.

5. Which of these exception is thrown by close() and read() methods?

- a) IOException**
- b) FileException**
- c) FileNotFoundException**
- d) FileInputOutputException**

Answer: a

Explanation: Both close() and read() method throw IOException.

6. Which of these methods is used to write() into a file?

- a) put()**
- b) putFile()**
- c) write()**
- d) writeFile()**

Answer: c

Explanation: None.

7. What will be the output of the following Java program?

```
1.      import java.io.*;
2.
3.      class filesinputoutput
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              InputStream obj = new FileInputStream("inputoutput.java");
10.
11.              System.out.print(obj.available());
12.
13.          }
14.      }
```

Note: inputoutput.java is stored in the disk.

- a) true**
- b) false**
- c) prints number of bytes in file**
- d) prints number of characters in the file**

Answer: c

Explanation: obj.available() returns the number of bytes.

Output:

```
$ javac filesinputoutput.java
$ java filesinputoutput
1422
```

(Output will be different in your case)

8. What will be the output of the following Java program?

```
1.  import java.io.*;
2.  public class filesinputoutput
3.  {
4.      public static void main(String[] args)
5.      {
6.          String obj  = "abc";
7.          byte b[] = obj.getBytes();
8.          ByteArrayInputStream obj1 = new ByteArrayInputStream(b);
9.          for (int i = 0; i < 2; ++ i)
10.         {
11.             int c;
12.             while((c = obj1.read()) != -1)
13.             {
14.                 if(i == 0)
15.                 {
16.                     System.out.print(Character.toUpperCase((char) c));
17.                     obj2.write(1);
18.                 }
19.             }
20.             System.out.print(obj2);
21.         }
22.     }
23. }
```

- a) AaBaCa**
- b) ABCaaa**
- c) AaaBaaCaa**
- d) AaBaaCaaa**

Answer: d

Explanation: None.

Output:

```
$ javac filesinputoutput.java
$ java filesinputoutput
AaBaaCaaa
```

9. What will be the output of the following Java program?

```
1.  import java.io.*;
2.
3.  class Chararrayinput
4.  {
5.
6.      public static void main(String[] args)
7.      {
8.
9.          String obj  = "abcdef";
10.         int length = obj.length();
11.         char c[] = new char[length];
12.         obj.getChars(0, length, c, 0);
13.         CharArrayReader input1 = new CharArrayReader(c);
14.         CharArrayReader input2 = new CharArrayReader(c, 0, 3);
15.         int i;
16.         try
17.         {
18.             while((i = input2.read()) != -1)
19.             {
20.                 System.out.print((char)i);
21.             }
22.         }
23.         catch (IOException e)
24.         {
25.             e.printStackTrace();
26.         }
27.     }
```

- a) abc
- b) abcd
- c) abcde
- d) abcdef

Answer: a

Explanation: None.

Output:

```
$ javac Chararrayinput.java
```

```
$ java Chararrayinput  
abc
```

10. What will be the output of the following Java program?

```
1.    import java.io.*;  
2.    class Chararrayinput  
3.    {  
4.        public static void main(String[] args)  
5.        {  
6.            String obj  = "abcdefgh";  
7.            int length = obj.length();  
8.            char c[] = new char[length];  
9.            obj.getChars(0, length, c, 0);  
10.           CharArrayReader input1 = new CharArrayReader(c);  
11.           CharArrayReader input2 = new CharArrayReader(c, 1, 4);  
12.           int i;  
13.           int j;  
14.           try  
15.           {  
16.               while((i = input1.read()) == (j = input2.read()))  
17.               {  
18.                   System.out.print((char)i);  
19.               }  
20.           }  
21.           catch (IOException e)  
22.           {  
23.               e.printStackTrace();  
24.           }  
25.       }  
26.   }
```

- a) abc
- b) abcd
- c) abcde
- d) none of the mentioned

Answer: d
Explanation: No output is printed. CharArrayReader object input1 contains string “abcdefgh” whereas object input2 contains string “bcde”, when while((i=input1.read())==(j=input2.read())) is executed the starting character of each

object is compared since they are unequal control comes out of loop and nothing is printed on the screen.
Output:

```
$ javac Chararrayinput.java  
$ java Chararrayinput
```

1. Which of these functions is called to display the output of an applet?

- a) display()
- b) paint()
- c) displayApplet()
- d) PrintApplet()

Answer: b

Explanation: Whenever the applet requires to redraw its output, it is done by using method paint().

2. Which of these methods can be used to output a string in an applet?

- a) display()
- b) print()
- c) drawString()
- d) transient()

Answer: c

Explanation: drawString() method is defined in Graphics class, it is used to output a string in an applet.

3. Which of these methods is a part of Abstract Window Toolkit (AWT) ?

- a) display()
- b) paint()
- c) drawString()
- d) transient()

Answer: b

Explanation: paint() is an abstract method defined in AWT.

4. Which of these modifiers can be used for a variable so that it can be accessed from any thread or parts of a program?

- a) transient
- b) volatile
- c) global
- d) No modifier is needed

Answer: b

Explanation: The volatile modifier tells the compiler that the variable modified by volatile can be changed unexpectedly by other part of the program. Specially used in situations involving multithreading.

5. Which of these operators can be used to get run time information about an object?

- a) getInfo
- b) Info
- c) instanceof
- d) getinfoof

Answer: c

Explanation: None.

6. What is the Message is displayed in the applet made by the following Java program?

```
1.      import java.awt.*;  
2.      import java.applet.*;
```

```
3.      public class myapplet extends Applet
4.      {
5.          public void paint(Graphics g)
6.          {
7.              g.drawString("A Simple Applet", 20, 20);
8.          }
9.      }
```

- a) A Simple Applet
- b) A Simple Applet 20 20
- c) Compilation Error
- d) Runtime Error

Answer: a

Explanation: None.

Output:

A Simple Applet

(Output comes in a new java application)

7. What is the length of the application box made by the following Java program?

```
1.      import java.awt.*;
2.      import java.applet.*;
3.      public class myapplet extends Applet
4.      {
5.          public void paint(Graphics g)
6.          {
7.              g.drawString("A Simple Applet", 20, 20);
8.          }
9.      }
```

- a) 20
- b) 50
- c) 100
- d) System dependent

Answer: a

Explanation: the code in pain() method – g.drawString(“A Simple Applet”,20,20); draws a applet box of length 20 and width 20.

8. What is the length of the application box made the following Java program?

```
1.      import java.awt.*;
2.      import java.applet.*;
3.      public class myapplet extends Applet
4.      {
```

```

5.      Graphic g;
6.      g.drawString("A Simple Applet", 20, 20);
7.  }

```

- a) 20
- b) Default value
- c) Compilation Error
- d) Runtime Error

Answer: c

Explanation: To implement the method `drawString` we need first need to define abstract method of AWT that is `paint()` method. Without `paint()` method we can not define and use `drawString` or any `Graphic` class methods.

9. What will be the output of the following Java program?

```

1.      import java.io.*;
2.
3.      class Chararrayinput
4.      {
5.
6.          public static void main(String[] args)
7.          {
8.
9.              String obj  = "abcdefgh";
10.             int length = obj.length();
11.             char c[] = new char[length];
12.             obj.getChars(0, length, c, 0);
13.             CharArrayReader input1 = new CharArrayReader(c);
14.             CharArrayReader input2 = new CharArrayReader(c, 1, 4);
15.
16.             int i;
17.             int j;
18.             try
19.             {
20.                 while((i = input1.read()) == (j = input2.read()))
21.                 {
22.                     System.out.print((char)i);
23.                 }
24.             }
25.             catch (IOException e)
26.             {
27.                 e.printStackTrace();
28.             }
29.         }
30.     }

```

- a) abc
- b) abcd
- c) abcde
- d) none of the mentioned

Answer: d

Explanation: No output is printed. CharArrayReader object input1 contains string "abcdefgh" whereas object input2 contains string "bcde", when while((i=input1.read())==(j=input2.read())) is executed the starting character of each object is compared since they are unequal control comes out of loop and nothing is printed on the screen.

Output:

```
$ javac Chararrayinput.java
$ java Chararrayinput
```

1. Which of these package is used for text formatting in Java programming language?

- a) java.text
- b) java.awt
- c) java.awt.text
- d) java.io

Answer: a

Explanation: java.text allows formatting, searching and manipulating text.

2. Which of this class can be used to format dates and times?

- a) Date
- b) SimpleDate
- c) DateFormat
- d) textFormat

Answer: c

Explanation: DateFormat is an abstract class that provides the ability to format and parse dates and times.

3. Which of these method returns an instance of DateFormat that can format time information?

- a) getTime()
- b) getTimeInstance()
- c) getTimeDateinstance()
- d) getDateFormatinstance()

Answer: b

Explanation: getTimeInstance() method returns an instance of DateFormat that can format time information.

4. Which of these class allows us to define our own formatting pattern for dates and time?

- a) DefinedDateFormat
- b) SimpleDateFormat
- c) ComplexDateFormat
- d) UsersDateFormat

Answer: b

Explanation: The SimpleDateFormat is a concrete subclass of DateFormat. It allows you to define your own formatting patterns that are used to display date and time information.

5. Which of these formatting strings of SimpleDateFormat class is used to print AM or PM in time?

- a) a
- b) b
- c) c
- d) d

Answer: a

Explanation: By using format string “a” we can print AM/PM in time.

6. Which of these formatting strings of SimpleDateFormat class is used to print week of the year?

- a) w**
- b) W**
- c) s**
- d) S**

Answer: a

Explanation: By using format string “w” we can print week in a year whereas by using ‘W’ we can print week of a month.

7. What will be the output of the following Java program?

```
1.    import java.text.*;
2.    import java.util.*;
3.    class Date_formatting
4.    {
5.        public static void main(String args[])
6.        {
7.            Date date = new Date();
8.            SimpleDateFormat sdf;
9.            sdf = new SimpleDateFormat("mm:hh:ss");
10.           System.out.print(sdf.format(date));
11.        }
12.    }
```

Note : The program is executed at 3 hour 55 minutes and 4 sec (24 hours time).

- a) 3:55:4**
- b) 3.55.4**
- c) 55:03:04**
- d) 03:55:04**

Answer: c

Explanation: None.

Output:

```
$ javac Date_formatting.java
$ java Date_formatting
55:03:04
```

8. What will be the output of the following Java program?

```
1.    import java.text.*;
2.    import java.util.*;
3.    class Date_formatting
4.    {
```



```

5.         public static void main(String args[])
6.         {
7.             Date date = new Date();
8.             SimpleDateFormat sdf;
9.             sdf = new SimpleDateFormat("hh:mm:ss");
10.            System.out.print(sdf.format(date));
11.        }
12.    }

```

Note : The program is executed at 3 hour 55 minutes and 4 sec (24 hours time).

- a) 3:55:4
- b) 3.55.4
- c) 55:03:04
- d) 03:55:04

Answer: d

Explanation: The code “sdf = new SimpleDateFormat(“hh:mm:ss”);” create a SimpleDateFormat class with format hh:mm:ss where h is hours, m is month and s is seconds.

Output:

```

$ javac Date_formatting.java
$ java Date_formatting
03:55:04

```

9. What will be the output of the following Java program?

```

1.     import java.text.*;
2.     import java.util.*;
3.     class Date_formatting
4.     {
5.         public static void main(String args[])
6.         {
7.             Date date = new Date();
8.             SimpleDateFormat sdf;
9.             sdf = new SimpleDateFormat("E MMM dd yyyy");
10.            System.out.print(sdf.format(date));
11.        }
12.    }

```

Note: The program is executed at 3 hour 55 minutes and 4 sec on Monday, 15 July(24 hours time).

- a) Mon Jul 15 2013
- b) Jul 15 2013
- c) 55:03:04 Mon Jul 15 2013
- d) 03:55:04 Jul 15 2013

Answer: a

Explanation: None.

Output:

```
$ javac Date_formatting.java
$ java Date_formatting
Mon Jul 15 2013
```

10. What will be the output of the following Java program?

```
1.      import java.text.*;
2.      import java.util.*;
3.      class Date_formatting
4.      {
5.          public static void main(String args[])
6.          {
7.              Date date = new Date();
8.              SimpleDateFormat sdf;
9.              sdf = new SimpleDateFormat("z");
10.             System.out.print(sdf.format(date));
11.         }
12.     }
```

Note : The program is executed at 3 hour 55 minutes and 4 sec on Monday, 15 July(24 hours time).

- a) z
- b) Jul
- c) Mon
- d) PDT

Answer: d

Explanation: format string “z” is used to print time zone.

Output:

```
$ javac Date_formatting.java
$ java Date_formatting
PDT
```

1. Which of the following is not a class of java.util.regex?

- a) Pattern class
- b) matcher class
- c) PatternSyntaxException
- d) Regex class

Answer: d

Explanation: java.util.regex consists 3 classes. PatternSyntaxException indicates syntax error in regex.

2. What is the significance of Matcher class for regular expression in java?

- a) interpretes pattern in the string
- b) Performs match in the string
- c) interpreted both pattern and performs match operations in the string
- d) None of the mentioned.

Answer: c

Explanation: matcher() method is invoked using matcher object which interpretes pattern and performs match operations in the input string.

3. Object of which class is used to compile regular expression?

- a) Pattern class**
- b) Matcher class**
- c) PatternSyntaxException**
- d) None of the mentioned**

Answer: a

Explanation: object of Pattern class can represent compiled regular expression.

4. Which capturing group can represent the entire expression?

- a) group ***
- b) group 0**
- c) group * or group 0**
- d) None of the mentioned**

Answer: b

Explanation: Group 0 is a special group which represents the entire expression.

Answer: a
Explanation: groupCount reports total number of Capturing groups. this does not include special group, group 0.

6. Which of the following matches nonword character using regular expression in java?

- a) \w**
- b) \W**
- c) \s**
- d) \S**

Answer: b

Explanation: \W matches nonword characters. [0-9], [A-Z] and _ (underscore) are word characters. All other than these characters are nonword characters.

7. Which of the following matches end of the string using regular expression in java?

- a) \z**
- b) **
- c) ***
- d) \Z**

Answer: a

Explanation: \z is used to match end of the entire string in regular expression in java.

8. What does public int end(int group) return?

- a) offset from last character of the subsequent group**
- b) offset from first character of the subsequent group**
- c) offset from last character matched**
- d) offset from first character matched**

Answer: a

Explanation: public int end(int group) returns offset from the last character of the subsequent group.

9. what does public String replaceAll(string replace) do?

- a) Replace all characters that matches pattern with a replacement string**
- b) Replace first subsequence that matches pattern with a replacement string**
- c) Replace all other than first subsequence of that matches pattern with a replacement string**
- d) Replace every subsequence of the input sequence that matches pattern with a replacement string**

Answer: d

Explanation: replaceAll method replaces every subsequence of the sequence that matches pattern with a replacement string.

- 10. What does public int start() return?**
- a) returns start index of the input string**
 - b) returns start index of the current match**
 - c) returns start index of the previous match**
 - d) none of the mentioned**

Answer: c

Explanation: public int start() returns index of the previous match in the input string.

- 1. Which of these packages contains all the classes and methods required for even handling in Java?**
- a) java.applet**
 - b) java.awt**
 - c) java.event**
 - d) java.awt.event**

Answer: d

Explanation: Most of the event to which an applet response is generated by a user. Hence they are in Abstract Window Kit package, java.awt.event.

- 2. What is an event in delegation event model used by Java programming language?**
- a) An event is an object that describes a state change in a source**
 - b) An event is an object that describes a state change in processing**
 - c) An event is an object that describes any change by the user and system**
 - d) An event is a class used for defining object, to create events**

Answer: a

Explanation: An event is an object that describes a state change in a source.

- 3. Which of these methods are used to register a keyboard event listener?**
- a) KeyListener()**
 - b) addKistener()**
 - c) addKeyListener()**
 - d) eventKeyboardListener()**

Answer: c

Explanation: None.

- 4. Which of these methods are used to register a mouse motion listener?**
- a) addMouse()**
 - b) addMouseListener()**
 - c) addMouseMotionListner()**
 - d) eventMouseMotionListener()**

Answer: c

Explanation: None.

- 5. What is a listener in context to event handling?**
- a) A listener is a variable that is notified when an event occurs**
 - b) A listener is a object that is notified when an event occurs**
 - c) A listener is a method that is notified when an event occurs**
 - d) None of the mentioned**

Answer: b

Explanation: A listener is a 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 notification about specific event types, and secondly it must

implement methods to receive and process these notifications.

6. Event class is defined in which of these libraries?

- a) java.io**
- b) java.lang**
- c) java.net**
- d) java.util**

Answer: d

Explanation: None.

7. Which of these methods can be used to determine the type of event?

- a) getID()**
- b) getSource()**
- c) getEvent()**
- d) getEventObject()**

Answer: a

Explanation: getID() can be used to determine the type of an event.

8. Which of these class is super class of all the events?

- a) EventObject**
- b) EventClass**
- c) ActionEvent**
- d) ItemEvent**

Answer: a

Explanation: EventObject class is a super class of all the events and is defined in java.util package.

9. Which of these events will be notified if scroll bar is manipulated?

- a) ActionEvent**
- b) ComponentEvent**
- c) AdjustmentEvent**
- d) WindowEvent**

Answer: c

Explanation: AdjustmentEvent is generated when a scroll bar is manipulated.

10. Which of these events will be generated if we close an applet's window?

- a) ActionEvent**
- b) ComponentEvent**
- c) AdjustmentEvent**
- d) WindowEvent**

Answer: d

Explanation: WindowEvent is generated when a window is activated, closed, deactivated, deiconfied, iconfied, opened or quit.

1. Which of these events is generated when a button is pressed?

- a) ActionEvent**
- b) KeyEvent**
- c) WindowEvent**
- d) AdjustmentEvent**

Answer: a

Explanation: Action event is generated when a button is pressed, a list item is double-clicked or a menu item is selected.

2. Which of these methods can be used to obtain the command name for invoking ActionEvent object?

- a) `getCommand()`
- b) `getActionCommand()`
- c) `getActionEvent()`
- d) `getActionEventCommand()`

Answer: b

Explanation: None.

3. Which of these are integer constants defined in `ActionEvent` class?

- a) `ALT_MASK`
- b) `CTRL_MASK`
- c) `SHIFT_MASK`
- d) All of the mentioned

Answer: d

Explanation: Action event defines 4 integer constants `ALT_MASK`, `CTRL_MASK`, `SHIFT_MASK` and `ACTION_PERFORMED`

4. Which of these methods can be used to know which key is pressed?

- a) `getKey()`
- b) `getModifier()`
- c) `getActionKey()`
- d) `getActionEvent()`

Answer: b

Explanation: The `getModifiers()` methods returns a value that indicates which modifiers keys (`ALT`, `CTRL`, `META`, `SHIFT`) were pressed when the event was generated.

5. Which of these events is generated by scroll bar?

- a) `ActionEvent`
- b) `KeyEvent`
- c) `WindowEvent`
- d) `AdjustmentEvent`

Answer: d

Explanation: None.

6. Which of these methods can be used to determine the type of adjustment event?

- a) `getType()`
- b) `getEventType()`
- c) `getAdjustmentType()`
- d) `getEventObjectType()`

Answer: c

Explanation: None.

7. Which of these methods can be used to know the degree of adjustment made by the user?

- a) `getValue()`
- b) `getAdjustmentType()`
- c) `getAdjustmentValue()`
- d) `getAdjustmentAmount()`

Answer: a

Explanation: The amount of the adjustment can be obtained from the `getValue()` method, it returns an integer value corresponding to the amount of adjustment made.

8. Which of these constant value will change when the button at the end of scroll bar was clicked to increase its value?

- a) **BLOCK_DECREMENT**
- b) **BLOCK_INCREMENT**
- c) **UNIT_DECREMENT**
- d) **UNIT_INCREMENT**

Answer: d

Explanation: UNIT_INCREMENT VALUE will change when the button at the end of scroll bar was clicked to increase its value.

1. Which of these events is generated when the size of an event is changed?

- a) **ComponentEvent**
- b) **ContainerEvent**
- c) **FocusEvent**
- d) **InputEvent**

Answer: a

Explanation: A ComponentEvent is generated when the size, position or visibility of a component is changed.

2. Which of these events is generated when the component is added or removed?

- a) **ComponentEvent**
- b) **ContainerEvent**
- c) **FocusEvent**
- d) **InputEvent**

Answer: b

Explanation: A ContainerEvent is generated when a component is added to or removed from a container. It has two integer constants COMPONENT_ADDED & COMPONENT_REMOVED.

3. Which of these methods can be used to obtain the reference to the container that generated a ContainerEvent?

- a) **getContainer()**
- b) **getContainerCommand()**
- c) **getActionEvent()**
- d) **getContainerEvent()**

Answer: d

Explanation: None.

4. Which of these methods can be used to get reference to a component that was removed from a container?

- a) **getComponent()**
- b) **getchild()**
- c) **getContainerComponent()**
- d) **getComponentChild()**

Answer: b

Explanation: The getChild() method returns a reference to the component that was added to or removed from the container.

5. Which of these are integer constants of ComponentEvent class?

- a) **COMPONENT_HIDDEN**
- b) **COMPONENT_MOVED**
- c) **COMPONENT_RESIZE**
- d) **All of the mentioned**

Answer: d

Explanation: The component event class defines 4 constants COMPONENT_HIDDEN, COMPONENT-MOVED, COMPONENT-RESIZE and COMPONENT-SHOWN.

6. Which of these events is generated when computer gains or loses input focus?

- a) **ComponentEvent**
- b) **ContainerEvent**
- c) **FocusEvent**
- d) **InputEvent**

Answer: c

Explanation: None.

7. FocusEvent is subclass of which of these classes?

- a) **ComponentEvent**
- b) **ContainerEvent**
- c) **ItemEvent**
- d) **InputEvent**

Answer: a

Explanation: None.

8. Which of these methods can be used to know the type of focus change?

- a) **typeFocus()**
- b) **typeEventFocus()**
- c) **isTemporary()**
- d) **isPermanent()**

Answer: c

Explanation: There are two types of focus events – permanent and temporary. The isTemporary() method indicates if this focus change is temporary, it returns a Boolean value.

9. Which of these is superclass of ContainerEvent class?

- a) **WindowEvent**
- b) **ComponentEvent**
- c) **ItemEvent**
- d) **InputEvent**

Answer: b

Explanation: ContainerEvent is superclass of ContainerEvent, FocusEvent, KeyEvent, MouseEvent and WindowEvent.

1. Which of these events is generated when the window is closed?

- a) **TextEvent**
- b) **MouseEvent**
- c) **FocusEvent**
- d) **WindowEvent**

Answer: d

Explanation: A WindowEvent is generated when a window is opened, close, activated or deactivated.

2. Which of these methods can be used to obtain the coordinates of a mouse?

- a) **getPoint()**
- b) **getCoordinates()**
- c) **getMouseXY()**
- d) **getMouseCordinates()**

Answer: a

Explanation: getPoint() method can be used to obtain coordinates of a mouse, alternatively we can use getX() and getY() methods for x and y coordinates of mouse respectively.

3. Which of these methods can be used to change location of an event?

- a) **ChangePoint()**
- b) **TranslatePoint()**

- c) **ChangeCoordinates()**
- d) **TranslateCoordinates()**

Answer: b

Explanation: None.

4. Which of these are integer constants of TextEvent class?

- a) **TEXT_CHANGED**
- b) **TEXT_FORMAT_CHANGED**
- c) **TEXT_VALUE_CHANGED**
- d) **TEXT_SIZE_CHANGED**

Answer: c

Explanation: TextEvent defines a single integer constant TEXT_VALUE_CHANGED.

5. Which of these methods is used to obtain the object that generated a WindowEvent?

- a) **getMethod()**
- b) **getWindow()**
- c) **getWindowEvent()**
- d) **getWindowObject()**

Answer: b

Explanation: None.

6. MouseEvent is subclass of which of these classes?

- a) **ComponentEvent**
- b) **ContainerEvent**
- c) **ItemEvent**
- d) **InputEvent**

Answer: d

Explanation: None.

7. Which of these methods is used to get x coordinate of the mouse?

- a) **getX()**
- b) **getXCoordinate()**
- c) **getCoordinateX()**
- d) **getPointX()**

Answer: a

Explanation: getX() and getY() are used to obtain X AND Y coordinates of the mouse.

8. Which of these are constants defined in WindowEvent class?

- a) **WINDOW_ACTIVATED**
- b) **WINDOW_CLOSED**
- c) **WINDOW_DEICONIFIED**
- d) **All of the mentioned**

Answer: d

Explanation: WindowEvent class defines 7 constants – WINDOW_ACTIVATED, WINDOW_CLOSED, WINDOW_OPENED, WINDOW_DECONIFIED, WINDOW_CLOSING, WINDOW_DEACTIVATED, WINDOW_ICONIFIED.

9. Which of these is superclass of WindowEvent class?

- a) **WindowEvent**
- b) **ComponentEvent**
- c) **ItemEvent**
- d) **InputEvent**

Answer: b

Explanation: ComponentEvent is superclass of ContainerEvent, FocusEvent, KeyEvent, MouseEvent and WindowEvent.

1. Which of these packages contains all the event handling interfaces?

- a) java.lang**
- b) java.awt**
- c) java.awt.event**
- d) java.event**

Answer: c

Explanation: None.

2. Which of these interfaces handles the event when a component is added to a container?

- a) ComponentListener**
- b) ContainerListener**
- c) FocusListener**
- d) InputListener**

Answer: b

Explanation: The ContainerListener defines methods to recognize when a component is added to or removed from a container.

3. Which of these interfaces define a method actionPerformed()?

- a) ComponentListener**
- b) ContainerListener**
- c) ActionListener**
- d) InputListener**

Answer: c

Explanation: ActionListener defines the actionPerformed() method that is invoked when an adjustment event occurs.

4. Which of these interfaces define four methods?

- a) ComponentListener**
- b) ContainerListener**
- c) ActionListener**
- d) InputListener**

Answer: a

Explanation: ComponentListener defines four methods componentResized(), componentMoved(), componentShown() and componentHidden().

5. Which of these interfaces define a method itemStateChanged()?

- a) ComponentListener**
- b) ContainerListener**
- c) ActionListener**
- d) ItemListener**

Answer: d

Explanation: None.

6. Which of these methods will respond when you click any button by mouse?

- a) mouseClicked()**
- b) mouseEntered()**
- c) mousePressed()**
- d) all of the mentioned**

Answer: d

Explanation: when we click a button, first we enter the region of button hence mouseEntered() method responds then we

press the button which leads to respond from mouseClicked() and mousePressed().

7. Which of these methods will be invoked if a character is entered?

- a) keyPressed()
- b) keyReleased()
- c) keyTyped()
- d) keyEntered()

Answer: c

Explanation: None.

8. Which of these methods is defined in MouseMotionAdapter class?

- a) mouseDragged()
- b) mousePressed()
- c) mouseReleased()
- d) mouseClicked()

Answer: a

Explanation: The MouseMotionAdapter class defines 2 methods – mouseDragged() and mouseMoved.

9. Which of these is a superclass of all Adapter classes?

- a) Applet
- b) ComponentEvent
- c) Event
- d) InputEvent

Answer: a

Explanation: All Adapter classes extend Applet class.

1. Which class is used to generate random number?

- a) java.lang.Object
- b) java.util.randomNumber
- c) java.util.Random
- d) java.util.Object

Answer: c

Explanation: java.util.random class is used to generate random numbers in java program.

2. Which method is used to generate boolean random values in java?

- a) nextBoolean()
- b) randomBoolean()
- c) previousBoolean()
- d) generateBoolean()

Answer: a

Explanation: nextBoolean() method of java.util.Random class is used to generate random numbers.

3. What is the return type of Math.random() method?

- a) Integer
- b) Double
- c) String
- d) Boolean

Answer: b

Explanation: Math.random() method returns floating point number or precisely a double.

Answer: b
Explanation: Random is not a final class and can be extended to implement the algorithm as per requirement.

5. What is the range of numbers returned by Math.random() method?

- a) -1.0 to 1.0
- b) -1 to 1
- c) 0 to 100
- d) 0.0 to 1.0

Answer: d

Explanation: Math.random() returns only double value greater than or equal to 0.0 and less than 1.0.

6. How many bits are used for generating random numbers?

- a) 32
- b) 64
- c) 48
- d) 8

Answer: c

Explanation: Random number can accept 64 bits but it only uses 48 bits for generating random numbers.

- a) Random number between 1 to 15, including 1 and 15
- b) Random number between 1 to 15, excluding 15
- c) Random number between 1 to 15, excluding 1
- d) Random number between 1 to 15, excluding 1 and 15

Answer: a

Explanation: random.nextInt(15) + 1; returns random numbers between 1 to 15 including 1 and 15.

- a) Random number between 4 to 7, including 4 and 7
- b) Random number between 4 to 7, excluding 4 and 7
- c) Random number between 4 to 10, excluding 4 and 10
- d) Random number between 4 to 10, including 4 and 10

Answer: d

Explanation: random.nextInt(7) + 4; returns random numbers between 4 to 10 including 4 and 10. it follows "nextInt(max - min + 1) + min" formula.

Answer: b
Explanation: Math.random() doesn't guarantee uniqueness. To guarantee uniqueness we must store the generated value in the database and compare against already generated values.

10. What is the signature of Math.random() method?

- a) public static double random()
- b) public void double random()
- c) public static int random()
- d) public void int random()

Answer: a

Explanation: public static double random() is the utility method provided by Math class which returns double.

1. Which of these class produce objects with respect to geographical locations?

- a) TimeZone
- b) Locale
- c) Date
- d) SimpleTimeZone

Answer: b

Explanation: The Locale class is instantiated to produce objects that each describe a geographical or cultural region.

2. Which of these methods is not a Locale class?

- a) UK
- b) US
- c) INDIA

d) KOREA

Answer: c

Explanation: INDIA is not a Locale class.

3. Which of these class can generate pseudorandom numbers?

- a) Locale**
- b) Rand**
- c) Random**
- d) None of the mentioned**

Answer: c

Explanation: None.

4. Which of these method of Locale class can be used to obtain country of operation?

- a) getCountry()**
- b) whichCountry()**
- c) DisplayCountry()**
- d) getDisplayCountry()**

Answer: d

Explanation: None.

5. Which of these is a method can generate a boolean output?

- a) retbool()**
- b) getBool()**
- c) nextBool()**
- d) nextBoolean()**

Answer: d

Explanation: None.

6. What will be the output of the following Java program?

```
1.      import java.util.*;
2.
3.      class LOCALE_CLASS
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              Locale obj = new Locale("INDIA") ;
10.
11.              System.out.print(obj.getCountry());
12.
13.          }
14.      }
```

- a) India**
- b) INDIA**
- c) Compilation Error**
- d) Nothing is displayed**

Answer: d

Explanation: None.

Output:

```
$ javac LOCALE_CLASS.java
$ java LOCALE_CLASS
```

7. What will be the output of the following Java program?

```
1.    import java.util.*;
2.
3.    class LOCALE_CLASS
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            Locale obj = new Locale("HINDI", "INDIA") ;
10.
11.            System.out.print(obj.getCountry());
12.
13.        }
14.    }
```

- a) India
- b) INDIA
- c) Compilation Error
- d) Nothing is displayed

Answer: b

Explanation: None.

Output:

```
$ javac LOCALE_CLASS.java
$ java LOCALE_CLASS
INDIA
```

8. What will be the output of the following Java program?

```
1.    import java.util.*;
2.
3.    class LOCALE_CLASS
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            Locale obj = new Locale("HINDI") ;
10.
11.            System.out.print(obj.getDisplayLanguage());
12.
13.        }
14.    }
```

- a) India
- b) INDIA
- c) HINDI
- d) Nothing is displayed

Answer: c

Explanation: None.

Output:

```
$ javac LOCALE_CLASS.java
$ java LOCALE_CLASS
HINDI
```

9. What will be the output of the following Java program?

```
1.    import java.util.*;
2.
3.    class LOCALE_CLASS
4.    {
5.
6.        public static void main(String args[])
7.        {
8.
9.            Locale obj = new Locale("HINDI", "INDIA") ;
10.
11.            System.out.print(obj.getDisplayLanguage()) ;
12.
13.        }
14.    }
```

- a) India
- b) INDIA
- c) HINDI
- d) Nothing is displayed

Answer: c

Explanation: None.

Output:

```
$ javac LOCALE_CLASS.java
$ java LOCALE_CLASS
HINDI
```

1. What is the use of Observable class?

- a) It is used to create global subclasses
- b) It is used to create classes that other part of the program can observe
- c) It is used to create classes that can be accessed by other parts of program
- d) It is used to create methods that can be accessed by other parts of program

Answer: b

Explanation: The Observable class is used to create subclasses that other part of program can observe.

2. Which of these methods is used to notify observer the change in observed object?

- a) update()
- b) notify()
- c) check()
- d) observed()

Answer: a

Explanation: None.

3. Which of these methods calls update() method?

- a) notify()
- b) observeObject()
- c) updateObserver()
- d) notifyObserver()

Answer: d

Explanation: notifyObserver() notifies all the observers of the invoking object that it has changed by calling update(). A null is passed as the second argument to update().

- 4. Which of these methods is called when observed object has changed?**
- a) setChanged()**
 - b) update()**
 - c) notifyObserver()**
 - d) all of the mentioned**

Answer: d
Explanation: None.

- 5. Which of these classes can schedule task for execution in future?**
- a) Thread**
 - b) Timer**
 - c) System**
 - d) Observer**

Answer: b
Explanation: Timer and TimerTask are the classes that support the ability to schedule tasks for execution at some future time.

- 6. Which of these interfaces is implemented by TimerTask class?**
- a) Runnable**
 - b) Thread**
 - c) Observer**
 - d) ThreadCount**

Answer: a
Explanation: None.

- 7. Which of these package provides the ability to read and write in Zip format?**
- a) java.lang**
 - b) java.io**
 - c) java.util.zip**
 - d) java.util.zar**

Answer: c
Explanation: None.

- 1. Which of these keywords is used to define packages in Java?**
- a) pkg**
 - b) Pkg**
 - c) package**
 - d) Package**

Answer: c
Explanation: None.

- 2. Which of these is a mechanism for naming and visibility control of a class and its content?**
- a) Object**
 - b) Packages**
 - c) Interfaces**
 - d) None of the Mentioned.**

Answer: b
Explanation: 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.

3. Which of this access specifies can be used for a class so that its members can be accessed by a different class in the same package?

- a) Public
- b) Protected
- c) No Modifier
- d) All of the mentioned

Answer: d

Explanation: Either we can use public, protected or we can name the class without any specifier.

4. Which of these access specifiers can be used for a class so that its members can be accessed by a different class in the different package?

- a) Public
- b) Protected
- c) Private
- d) No Modifier

Answer: a

Explanation: None.

5. Which of the following is the correct way of importing an entire package ‘pkg’?

- a) import pkg.
- b) Import pkg.
- c) import pkg.*
- d) Import pkg.*

Answer: c

Explanation: Operator * is used to import the entire package.

6. Which of the following is an incorrect statement about packages?

- a) Package defines a namespace in which classes are stored
- b) A package can contain other package within it
- c) Java uses file system directories to store packages
- d) A package can be renamed without renaming the directory in which the classes are stored

Answer: d

Explanation: A package can be renamed only after renaming the directory in which the classes are stored.

7. Which of the following package stores all the standard java classes?

- a) lang
- b) java
- c) util
- d) java.packages

Answer: b

Explanation: None.

8. What will be the output of the following Java program?

```
1. package pkg;
2. class display
3. {
4.     int x;
5.     void show()
6.     {
```

```

7.         if (x > 1)
8.             System.out.print(x + " ");
9.     }
10. }
11. class packages
12. {
13.     public static void main(String args[])
14.     {
15.         display[] arr=new display[3];
16.         for(int i=0;i<3;i++)
17.             arr[i]=new display();
18.         arr[0].x = 0;
19.         arr[1].x = 1;
20.         arr[2].x = 2;
21.         for (int i = 0; i < 3; ++i)
22.             arr[i].show();
23.     }
24. }

```

Note : packages.class file is in directory pkg;

- a) 0
- b) 1
- c) 2
- d) 0 1 2

Answer: c

Explanation: None.

Output:

```

$ javac packages.java
$ java packages
2

```

9. What will be the output of the following Java program?

```

1.     package pkg;
2.
3.     class output
4.     {
5.         public static void main(String args[])
6.         {
7.             StringBuffer s1 = new StringBuffer("Hello");

```

```
8.         System.out.println(s1);
9.     }
10. }
```

- a) xello
- b) xxxxx
- c) Hxlllo
- d) Hexlo

Answer: c

Explanation: None.

Output:

```
$ javac output.java
$ java output
Hxlllo
```

10. What will be the output of the following Java program?

```
1.     package pkg;
2.
3.     class output
4.     {
5.
6.         public static void main(String args[])
7.         {
8.
9.             StringBuffer s1 = new StringBuffer("Hello World");
10.
11.             s1.insert(6 , "Good ");
12.
13.             System.out.println(s1);
14.
15.         }
16.     }
```

Note : Output.class file is not in directory pkg.

- a) HelloGoodWorld
- b) HellGoodoWorld
- c) Compilation error
- d) Runtime error

Answer: d

Explanation: Since output.class file is not in the directory pkg in which class output is defined, program will not be able to run.

output:

```
$ javac output.java
$ java output
can not find file output.class
```

1. Which of these keywords is used to define interfaces in Java?

- a) interface
- b) Interface
- c) intf
- d) Intf

Answer: a

Explanation: None.

2. Which of these can be used to fully abstract a class from its implementation?

- a) Objects**
- b) Packages**
- c) Interfaces**
- d) None of the Mentioned**

Answer: c

Explanation: None.

3. Which of these access specifiers can be used for an interface?

- a) Public**
- b) Protected**
- c) private**
- d) All of the mentioned**

Answer: a

Explanation: Access specifier of an interface is either public or no specifier. When no access specifier is used then default access specifier is used due to which interface is available only to other members of the package in which it is declared, when declared public it can be used by any code.

4. Which of these keywords is used by a class to use an interface defined previously?

- a) import**
- b) Import**
- c) implements**
- d) Implements**

Answer: c

Explanation: interface is inherited by a class using implements.

5. Which of the following is the correct way of implementing an interface salary by class manager?

- a) class manager extends salary {}**
- b) class manager implements salary {}**
- c) class manager imports salary {}**
- d) none of the mentioned**

Answer: b

Explanation: None.

6. Which of the following is an incorrect statement about packages?

- a) Interfaces specifies what class must do but not how it does**
- b) Interfaces are specified public if they are to be accessed by any code in the program**
- c) All variables in interface are implicitly final and static**
- d) All variables are static and methods are public if interface is defined pubic**

Answer: d

Explanation: All methods and variables are implicitly public if interface is declared public.

7. What will be the output of the following Java program?

```
1.      interface calculate
2.      {
3.          void cal(int item);
4.      }
```

```

5.      class display implements calculate
6.      {
7.          int x;
8.          public void cal(int item)
9.          {
10.             x = item * item;
11.          }
12.      }
13.      class interfaces
14.      {
15.          public static void main(String args[])
16.          {
17.              display arr = new display;
18.              arr.x = 0;
19.              arr.cal(2);
20.              System.out.print(arr.x);
21.          }
22.      }

```

- a) 0
- b) 2
- c) 4
- d) None of the mentioned

Answer: c

Explanation: None.

Output:

```

$ javac interfaces.java
$ java interfaces
4

```

8. What will be the output of the following Java program?

```

1.      interface calculate
2.      {
3.          void cal(int item);
4.      }
5.      class displayA implements calculate
6.      {
7.          int x;

```

```

8.         public void cal(int item)
9.     {
10.         x = item * item;
11.     }
12. }
13. class displayB implements calculate
14. {
15.     int x;
16.     public void cal(int item)
17.     {
18.         x = item / item;
19.     }
20. }
21. class interfaces
22. {
23.     public static void main(String args[])
24.     {
25.         displayA arr1 = new displayA;
26.         displayB arr2 = new displayB;
27.         arr1.x = 0;
28.         arr2.x = 0;
29.         arr1.cal(2);
30.         arr2.cal(2);
31.         System.out.print(arr1.x + " " + arr2.x);
32.     }
33. }

```

- a) 0 0
- b) 2 2
- c) 4 1
- d) 1 4

Answer: c

Explanation: class displayA implements the interface calculate by doubling the value of item, where as class displayB implements the interface by dividing item by item, therefore variable x of class displayA stores 4 and variable x of class displayB stores 1.

Output:

```

$ javac interfaces.java
$ java interfaces

```

9. What will be the output of the following Java program?

```
1. interface calculate
2. {
3.     int VAR = 0;
4.     void cal(int item);
5. }
6. class display implements calculate
7. {
8.     int x;
9.     public void cal(int item)
10.    {
11.        if (item<2)
12.            x = VAR;
13.        else
14.            x = item * item;
15.    }
16. }
17. class interfaces
18. {
19.
20.     public static void main(String args[])
21.     {
22.         display[] arr=new display[3];
23.
24.         for(int i=0;i<3;i++)
25.             arr[i]=new display();
26.         arr[0].cal(0);
27.         arr[1].cal(1);
28.         arr[2].cal(2);
29.         System.out.print(arr[0].x+" " + arr[1].x + " " + arr[2].x);
30.     }
31. }
```

a) 0 1 2

- b) 0 2 4
- c) 0 0 4
- d) 0 1 4

Answer: c

Explanation: None.

output:

```
$ javac interfaces.java
$ java interfaces
0 0 4
```

1. Which of the following access specifiers can be used for an interface?

- a) Protected
- b) Private
- c) Public
- d) Public, protected, private

Answer: a

Explanation: Interface can have either public access specifier or no specifier. The reason is they need to be implemented by other classes.

2. Which of the following is the correct way of implementing an interface A by class B?

- a) class B extends A{}
- b) class B implements A{}
- c) class B imports A{}
- d) None of the mentioned

Answer: b

Explanation: Concrete class implements an interface. They can be instantiated.

Answer: a
Explanation: Concrete classes must implement all methods in an interface. Through interface multiple inheritance is possible.

4. What type of variable can be defined in an interface?

- a) public static
- b) private final
- c) public final
- d) static final

Answer: d

Explanation: variable defined in an interface is implicitly final and static. They are usually written in capital letters.

5. What does an interface contain?

- a) Method definition
- b) Method declaration
- c) Method declaration and definition
- d) Method name

Answer: b

Explanation: Interface contains the only declaration of the method.

6. What type of methods an interface contain by default?

- a) abstract
- b) static
- c) final
- d) private

Answer: a

Explanation: By default, interface contains abstract methods. The abstract methods need to be implemented by concrete classes.

7. What will happen if we provide concrete implementation of method in interface?

- a) The concrete class implementing that method need not provide implementation of that method**
- b) Runtime exception is thrown**
- c) Compilation failure**
- d) Method not found exception is thrown**

Answer: c

Explanation: The methods of interfaces are always abstract. They provide only method definition.

8. What happens when a constructor is defined for an interface?

- a) Compilation failure**
- b) Runtime Exception**
- c) The interface compiles successfully**
- d) The implementing class will throw exception**

Answer: a

Explanation: Constructor is not provided by interface as objects cannot be instantiated.

9. What happens when we access the same variable defined in two interfaces implemented by the same class?

- a) Compilation failure**
- b) Runtime Exception**
- c) The JVM is not able to identify the correct variable**
- d) The interfaceName.variableName needs to be defined**

Answer: d

Explanation: The JVM needs to distinctly know which value of variable it needs to use. To avoid confusion to the JVM interfaceName.variableName is mandatory.

Answer: b
Explanation: No, Constructor, Static Initialization Block, Instance Initialization Block and variables cannot be abstract.

1. Which of these package is used for graphical user interface?

- a) java.applet**
- b) java.awt**
- c) java.awt.image**
- d) java.io**

Answer: b

Explanation: java.awt provides capabilities for graphical user interface.

2. Which of this package is used for analyzing code during run-time?

- a) java.applet**
- b) java.awt**
- c) java.io**
- d) java.lang.reflect**

Answer: d

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

3. Which of this package is used for handling security related issues in a program?

- a) java.security**
- b) java.lang.security**
- c) java.awt.image**
- d) java.io.security**

Answer: a

Explanation: java.security handles certificates, keys, digests, signatures, and other security functions.

4. Which of these class allows us to get real time data about private and protected member of a class?

- a) java.io
- b) GetInformation
- c) ReflectPermission
- d) MembersPermission

Answer: c

Explanation: The ReflectPermission class allows reflection of private or protected members of a class. This was added after java 2.0 .

5. Which of this package is used for invoking a method remotely?

- a) java.rmi
- b) java.awt
- c) java.util
- d) java.applet

Answer: a

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

6. What will be the output of the following Java program?

```
1.      import java.lang.reflect.*;
2.
3.      class Additional_packages
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              Class c = Class.forName("java.awt.Dimension");
10.             Constructor constructors[] = c.getConstructors();
11.             for (int i = 0; i < constructors.length; i++)
12.                 System.out.println(constructors[i]);
13.             }
14.         catch (Exception e)
15.         {
16.             System.out.print("Exception");
17.         }
18.     }
```

- a) Program prints all the constructors of 'java.awt.Dimension' package
- b) Program prints all the possible constructors of class 'Class'
- c) Program prints "Exception"
- d) Runtime Error

Answer: a

Explanation: None.

Output:

```
$ javac Additional_packages.java
$ java Additional_packages
public java.awt.Dimension(java.awt.Dimension)
public java.awt.Dimension()
public java.awt.Dimension(int,int)
```

7. What will be the output of the following Java program?

```
1.      import java.lang.reflect.*;
2.
3.      class Additional_packages
4.      {
5.
6.          public static void main(String args[])
7.          {
8.
9.              Class c = Class.forName("java.awt.Dimension");
10.             Field fields[] = c.getFields();
11.             for (int i = 0; i < fields.length; i++)
12.                 System.out.println(fields[i]);
13.             }
14.         catch (Exception e)
15.         {
16.             System.out.print("Exception");
17.         }
18.     }
```

- a) Program prints all the constructors of ‘java.awt.Dimension’ package
- b) Program prints all the methods of ‘java.awt.Dimension’ package
- c) Program prints all the data members of ‘java.awt.Dimension’ package
- d) program prints all the methods and data member of ‘java.awt.Dimension’ package

Answer: c

Explanation: None.

Output:

```
$ javac Additional_packages.java
$ java Additional_packages
public int java.awt.Dimension.width
public int java.awt.Dimension.height
```

8. What is the length of the application box made in the following Java program?

```
1.      import java.awt.*;
```

```

2.     import java.applet.*;
3.
4.     public class myapplet extends Applet
5.     {
6.         Graphic g;
7.         g.drawString("A Simple Applet",20,20);
8.     }

```

a) 20

b) Default value

c) Compilation Error

d) Runtime Error

Answer: c

Explanation: To implement the method drawString we need first need to define abstract method of AWT that is paint() method. Without paint() method we cannot define and use drawString or any Graphic class methods.

9. What will be the output of the following Java program?

```

1.     import java.lang.reflect.*;
2.
3.     class Additional_packages
4.     {
5.         public static void main(String args[])
6.         {
7.             try
8.             {
9.                 Class c = Class.forName("java.awt.Dimension");
10.                Method methods[] = c.getMethods();
11.                for (int i = 0; i < methods.length; i++)
12.                    System.out.println(methods[i]);
13.            }
14.            catch (Exception e)
15.            {
16.                System.out.print("Exception");
17.            }
18.        }

```

a) Program prints all the constructors of 'java.awt.Dimension' package

b) Program prints all the methods of 'java.awt.Dimension' package

c) Program prints all the data members of 'java.awt.Dimension' package

d) program prints all the methods and data member of 'java.awt.Dimension' package

Answer: b

Explanation: None.

Output:

```
$ javac Additional_packages.java
$ java Additional_packages
public int java.awt.Dimension.hashCode()
public boolean java.awt.Dimension.equals(java.lang.Object)
public java.lang.String java.awt.Dimension.toString()
public java.awt.Dimension java.awt.Dimension.getSize()
public void java.awt.Dimension.setSize(double,double)
public void java.awt.Dimension.setSize(int,int)
public void java.awt.Dimension.setSize(java.awt.Dimension)
public double java.awt.Dimension.getHeight()
public double java.awt.Dimension.getWidth()
public java.lang.Object java.awt.geom.Dimension2D.clone()
public void java.awt.geom.Dimension2D.setSize(java.awt.geom.Dimension2D)
public final native java.lang.Class java.lang.Object.getClass()
public final native void java.lang.Object.notify()
public final native void java.lang.Object.notifyAll()
public final native void java.lang.Object.wait(long)
public final void java.lang.Object.wait(long,int)
public final void java.lang.Object.wait()
```

1. Why are generics used?

- a) Generics make code more fast
- b) Generics make code more optimised and readable
- c) Generics add stability to your code by making more of your bugs detectable at compile time
- d) Generics add stability to your code by making more of your bugs detectable at runtime

Answer: c

Explanation: Generics add stability to your code by making more of your bugs detectable at compile time.

2. Which of these type parameters is used for a generic class to return and accept any type of object?

- a) K
- b) N
- c) T
- d) V

Answer: c

Explanation: T is used for type, A type variable can be any non-primitive type you specify: any class type, any interface type, any array type, or even another type variable.

3. Which of these type parameters is used for a generic class to return and accept a number?

- a) K
- b) N
- c) T
- d) V

Answer: b

Explanation: N is used for Number.

4. Which of these is an correct way of defining generic class?

- a) class name(T1, T2, ..., Tn) { /* ... */ }
- b) class name<T1, T2, ..., Tn> { /* ... */ }
- c) class name[T1, T2, ..., Tn] { /* ... */ }
- d) class name{T1, T2, ..., Tn} { /* ... */ }

Answer: b

Explanation: The type parameter section, delimited by angle brackets (<>), follows the class name. It specifies the type parameters (also called type variables) T1, T2, ..., and Tn.

5. Which of the following is an incorrect statement regarding the use of generics and parameterized types in Java?
- a) Generics provide type safety by shifting more type checking responsibilities to the compiler
 - b) Generics and parameterized types eliminate the need for down casts when using Java Collections
 - c) When designing your own collections class (say, a linked list), generics and parameterized types allow you to achieve type safety with just a single class definition as opposed to defining multiple classes
 - d) All of the mentioned

Answer: c

Explanation: None.

6. Which of the following reference types cannot be generic?

- a) Anonymous inner class
- b) Interface
- c) Inner class
- d) All of the mentioned

Answer: a

Explanation: None.

7. What will be the output of the following Java program?

```
1.      public class BoxDemo
2.      {
3.          public static <U> void addBox(U u, java.util.List<Box<U>> boxes)
4.          {
5.              Box<U> box = new Box<>();
6.              box.set(u);
7.              boxes.add(box);
8.          }
9.          public static <U> void outputBoxes(java.util.List<Box<U>> boxes)
10.         {
11.             int counter = 0;
12.             for (Box<U> box: boxes)
13.             {
14.                 U boxContents = box.get();
15.                 System.out.println("Box #" + counter + " contains [" + boxContents.toString() + "]");
16.                 counter++;
17.             }
18.         }
19.         public static void main(String[] args)
20.         {
21.             java.util.ArrayList<Box<Integer>> listOfIntegerBoxes = new java.util.ArrayList<>();
22.             BoxDemo.<Integer>addBox(Integer.valueOf(10), listOfIntegerBoxes);
```

```
23.         BoxDemo.outputBoxes(listOfIntegerBoxes);
24.     }
25. }
```

- a) 10
- b) Box #0 [10]
- c) Box contains [10]
- d) Box #0 contains [10]

Answer: d
Explanation: None.
Output:

```
$ javac Output.javadoc
$ java Output
Box #0 contains [10].
```

8. What will be the output of the following Java program?

```
1.     public class BoxDemo
2.     {
3.         public static <U> void addBox(U u,
4.         java.util.List<Box<U>> boxes)
5.         {
6.             Box<U> box = new Box<>();
7.             box.set(u);
8.             boxes.add(box);
9.         }
10.        public static <U> void outputBoxes(java.util.List<Box<U>> boxes)
11.        {
12.            int counter = 0;
13.            for (Box<U> box: boxes)
14.            {
15.                U boxContents = box.get();
16.                System.out.println "[" + boxContents.toString() + " ]");
17.                counter++;
18.            }
19.        }
20.        public static void main(String[] args)
21.        {
22.            java.util.ArrayList<Box<Integer>> listOfIntegerBoxes = new java.util.ArrayList<>()
```

```

23.         BoxDemo.<Integer>addBox(Integer.valueOf(0), listOfIntegerBoxes);

24.         BoxDemo.outputBoxes(listOfIntegerBoxes);

25.     }

26. }

```

- a) 0
- b) 1
- c) [1]
- d) [0]

Answer: d

Explanation: None.

Output:

```

$ javac Output.javac
$ java Output
[0]

```

9. What will be the output of the following Java program?

```

1.     import java.util.*;

2.     public class genericstack <E>

3.     {

4.         Stack <E> stk = new Stack <E>();

5.         public void push(E obj)

6.         {

7.             stk.push(obj);

8.         }

9.         public E pop()

10.        {

11.            E obj = stk.pop();

12.            return obj;

13.        }

14.    }

15.    class Output

16.    {

17.        public static void main(String args[])

18.        {

19.            genericstack <String> gs = new genericstack<String>();

20.            gs.push("Hello");

21.            System.out.print(gs.pop() + " ");

```



```

22.         genericstack <Integer> gs = new genericstack<Integer>();
23.
24.         gs.push(36);
25.
26.         System.out.println(gs.pop());
27.     }
28. }

```

- a) Error
- b) Hello
- c) 36
- d) Hello 36

Answer: d
Explanation: None.
Output:

```

$ javac Output.javac
$ java Output
Hello 36

```

10. What will be the output of the following Java program?

```

1.     public class BoxDemo
2.     {
3.
4.         public static <U> void addBox(U u,
5.         java.util.List<Box<U>> boxes)
6.         {
7.             Box<U> box = new Box<>();
8.             box.set(u);
9.             boxes.add(box);
10.        }
11.
12.        public static <U> void outputBoxes(java.util.List<Box<U>> boxes)
13.        {
14.
15.            int counter = 0;
16.
17.            for (Box<U> box: boxes)
18.            {
19.
20.                U boxContents = box.get();
21.
22.                System.out.println("Box #" + counter + " contains [" + boxContents.toString()
23.
24.                counter++;
25.            }
26.        }
27.
28.        public static void main(String[] args)

```

```

21.         {
22.             java.util.ArrayList<Box<Integer>> listOfIntegerBoxes = new java.util.ArrayList<>()
23.             BoxDemo.<Integer>addBox(Integer.valueOf(10), listOfIntegerBoxes);
24.             BoxDemo.outputBoxes(listOfIntegerBoxes);
25.         }
26.     }

```

- a) 10
- b) Box #0 [10]
- c) Box contains [10]
- d) Box #0 contains [10]

Answer: d
 Explanation: None.
 Output:

```

$ javac Output.java
$ java Output
Box #0 contains [10].

```

1. JUnits are used for which type of testing?
- a) Unit Testing
 - b) Integration Testing
 - c) System Testing
 - d) Blackbox Testing

Answer: a
 Explanation: JUnit is a testing framework for unit testing. It uses java as a programming platform. It is managed by junit.org community.

2. Which of the below statement about JUnit is false?
- a) It is an open source framework
 - b) It provides an annotation to identify test methods
 - c) It provides test runners for running test
 - d) They cannot be run automatically

Answer: d
 Explanation: JUnits test can be run automatically and they check their own results and provide immediate feedback.

3. Which of the below is an incorrect annotation with respect to JUnits?
- a) @Test
 - b) @BeforeClass
 - c) @JUnit
 - d) @AfterEach

Answer: c
 Explanation: @Test is used to annotate method under test, @BeforeEach and @AfterEach are called before and after each method respectively. @BeforeClass and @AfterClass are called only once for each class.

4. Which of these is not a mocking framework?
- a) EasyMock
 - b) Mockito
 - c) PowerMock
 - d) MockJava

Answer: d
Explanation: EasyMock, jMock, Mockito, Unitils Mock, PowerMock and JMockit are a various mocking framework.

5. Which method is used to verify the actual and expected results in Junits?

- a) `assert()`
- b) `equals()`
- c) `==`
- d) `isEqual()`

Answer: a
Explanation: assert method is used to compare actual and expected results in Junit. It has various implementation like assertEquals, assertEqualsArray, assertEqualsFalse, assertEqualsNotNull, etc.

6. What does `assertSame()` method use for assertion?

- a) `equals()` method
- b) `isEqual()` method
- c) `==`
- d) `compare()` method

Answer: c
Explanation: == is used to compare the objects not the content. assertEquals() method compares to check if actual and expected are the same objects. It does not compare their content.

7. How to let junits know that they need to be run using PowerMock?

- a) `@PowerMock`
- b) `@RunWith(PowerMock)`
- c) `@RunWith(Junits)`
- d) `@RunWith(PowerMockRunner.class)`

Answer: d
Explanation: @RunWith(PowerMockRunner.class) signifies to use PowerMock JUnit runner. Along with that @PrepareForTest(User.class) is used to declare the class being tested. mockStatic(Resource.class) is used to mock the static methods.

8. How can we simulate if then behavior in Junits?

- a) `if{..} else{..}`
- b) `if(..){..} else{..}`
- c) `Mockito.when(...).thenReturn(...);`
- d) `Mockito.if(..).then(..);`

Answer: c
Explanation: Mockito.when(mockList.size()).thenReturn(100); assertEquals(100, mockList.size()); is the usage to implement if and then behavior.

9. What is used to inject mock fields into the tested object automatically?

- a) `@InjectMocks`
- b) `@Inject`
- c) `@InjectMockObject`
- d) `@Mock`

Answer: a
Explanation: @InjectMocks annotation is used to inject mock fields into the tested object automatically.

```
@InjectMocks
MyDictionary dic = new MyDictionary();
```

b)

1. <dependency>

```
2.     <groupId>junit</groupId>
3.     <artifactId>junit</artifactId>
4.     <version>4.8.1</version>
5. </dependency>
```

c)

```
1. <dependency>
2.     <groupId>org.junit</groupId>
3.     <artifactId>junit</artifactId>
4.     <version>4.8.1</version>
5. </dependency>
```

d)

```
1. <dependency>
2.     <groupId>mock.junit</groupId>
3.     <artifactId>junit</artifactId>
4.     <version>4.8.1</version>
5. </dependency>
```

1. Which of the following is not introduced with Java 8?

a) Stream API

b) Serialization

c) Spliterator

d) Lambda Expression

Answer: b

Explanation: Serialization is not introduced with Java 8. It was introduced with an earlier version of Java.

2. What is the purpose of BooleanSupplier function interface?

a) represents supplier of Boolean-valued results

b) returns Boolean-valued result

c) There is no such function interface

d) returns null if Boolean is passed as argument

Answer: a

Explanation: BooleanSupplier function interface represents supplier of Boolean-valued results.

3. What is the return type of lambda expression?

a) String

b) Object

c) void

d) Function

Answer: d

Explanation: Lambda expression enables us to pass functionality as an argument to another method, such as what action should be taken when someone clicks a button.

4. Which is the new method introduced in java 8 to iterate over a collection?

- a) `for (String i : StringList)`
- b) `foreach (String i : StringList)`
- c) `StringList.forEach()`
- d) `List.for()`

Answer: c

Explanation: Traversing through `forEach` method of `Iterable` with anonymous class.

```
1. StringList.forEach(new Consumer<Integer> ()
2. {
3.     public void accept(Integer t)
4.     {
5.     }
6. });
7. //Traversing with Consumer interface implementation
8. MyConsumer action = new MyConsumer();
9. StringList.forEach(action);
10.
11. }
```

5. What are the two types of Streams offered by java 8?

- a) sequential and parallel
- b) sequential and random
- c) parallel and random
- d) random and synchronized

Answer: a

Explanation: Sequential stream and parallel stream are two types of stream provided by java.

```
1. Stream<Integer> sequentialStream = myList.stream();
2. Stream<Integer> parallelStream = myList.parallelStream();
```

6. Which feature of java 8 enables us to create a work stealing thread pool using all available processors at its target?

- a) `workPool`
- b) `newWorkStealingPool`
- c) `threadPool`
- d) `workThreadPool`

Answer: b

Explanation: Executors `newWorkStealingPool()` method to create a work-stealing thread pool using all available processors as its target parallelism level.

7. What does `Files.lines(Path path)` do?

- a) It reads all the files at the path specified as a `String`
- b) It reads all the lines from a file as a `Stream`
- c) It reads the filenames at the path specified
- d) It counts the number of lines for files at the path specified

Answer: b

Explanation: Files.lines(Path path) that reads all lines from a file as a Stream.

8. What is Optional object used for?

- a) Optional is used for optional runtime argument
- b) Optional is used for optional spring profile
- c) Optional is used to represent null with absent value
- d) Optional means it's not mandatory for method to return object

Answer: c

Explanation: Optional object is used to represent null with absent value. This class has various utility methods to facilitate code to handle values as 'available' or 'not available' instead of checking null values.

9. What is the substitute of Rhino javascript engine in Java 8?

- a) Nashorn
- b) V8
- c) Inscript
- d) Narcissus

Answer: a

Explanation: Nashorn provides 2 to 10 times faster in terms of performance, as it directly compiles the code in memory and passes the bytecode to JVM. Nashorn uses invoke dynamic feature.

10. What does SAM stand for in the context of Functional Interface?

- a) Single Ambivalue Method
- b) Single Abstract Method
- c) Simple Active Markup
- d) Simple Abstract Markup

Answer: b

Explanation: SAM Interface stands for Single Abstract Method Interface. Functional Interface is also known as SAM Interface because it contains only one abstract method.

1. Which method is used to create a directory with fileattributes?

- a) Path.create()
- b) Path.createDirectory()
- c) Files.createDirectory(path, fileAttributes)
- d) Files.create(fileAttributes)

Answer: c

Explanation: New directory can be created using Files.createDirectory(path, fileAttribute).

2. Which method can be used to check fileAccessiblity?

- a) isReadable(path)
- b) isWritable(path)
- c) isExecutable(path)
- d) isReadable(path), isWritable(path), and isExecutable(path)

Answer: d

Explanation: File accessiblity can be checked using isReadable(Path), isWritable(Path), and isExecutable(Path).

3. How can we delete all files in a directory?

- a) Files.delete(path)
- b) Files.deleteDir()
- c) Directory.delete()
- d) Directory.delete(path)

Answer: a

Explanation: The delete(Path) method deletes the file or throws an exception if the deletion fails. If file does not exist a

NoSuchFileException is thrown.

4. How to copy the file from one location to other?

- a) **Files.copy(source, target)**
- b) **Path.copy(source, target)**
- c) **source.copy(target)**
- d) **Files.createCopy(target)**

Answer: a

Explanation: Files.copy(source, target) is used to copy a file from one location to another. There are various options available like REPLACE_EXISTING, COPY_ATTRIBUTES and NOFOLLOW_LINKS.

5. How can we get the size of specified file?

- a) **capacity(path)**
- b) **size(path)**
- c) **length(path)**
- d) **Path.size()**

Answer: b

Explanation: size(Path) returns the size of the specified file in bytes.

6. How to read entire file in one line using java 8?

- a) **Files.readAllLines()**
- b) **Files.read()**
- c) **Files.readFile()**
- d) **Files.lines()**

Answer: a

Explanation: Java 8 provides Files.readAllLines() which allows us to read entire file in one task. We do not need to worry about readers and writers.

7. How can we create a symbolic link to file?

- a) **createLink()**
- b) **createSymLink()**
- c) **createSymbolicLink()**
- d) **createTempLink()**

Answer: c

Explanation: createSymbolicLink() creates a symbolic link to a target.

8. How can we filter lines based on content?

- a) **lines.filter()**
- b) **filter(lines)**
- c) **lines.contains(filter)**
- d) **lines.select()**

Answer: a

Explanation: lines.filter(line -> line.contains("===> Loaded package")) can be used to filter out.

9. Which jar provides FileUtils which contains methods for file operations?

- a) **file**
- b) **apache commons**
- c) **file commons**
- d) **dir**

Answer: b

Explanation: FileUtils is a part of apache commons which provides various methods for file operations like writeStringToFile.

10. Which feature of java 7 allows to not explicitly close IO resource?

- a) try catch finally**
- b) IOException**
- c) AutoCloseable**
- d) Streams**

Answer: c

Explanation: Any class that has implemented Autocloseable releases the I/O resources.

1. Which of the following is not a core interface of Hibernate?

- a) Configuration**
- b) Criteria**
- c) SessionManagement**
- d) Session**

Answer: c

Explanation: SessionManagement is not a core interface of Hibernate. Configuration, Criteria, SessionFactory, Session, Query and Transaction are the core interfaces of Hibernate.

Answer: a
Explanation: SessionFactory is a thread-safe object. Multiple threads can access it simultaneously.

3. Which of the following methods returns proxy object?

- a) loadDatabase()**
- b) getDatabase()**
- c) load()**
- d) get()**

Answer: c

Explanation: load() method returns proxy object. load() method should be used if it is sure that instance exists.

4. Which of the following methods hits database always?

- a) load()**
- b) loadDatabase()**
- c) getDatabase()**
- d) get()**

Answer: d

Explanation: get() method hits database always. Also, get() method does not return proxy object.

5. Which of the following method is used inside session only?

- a) merge()**
- b) update()**
- c) end()**
- d) kill()**

Answer: b

Explanation: update() method can only be used inside session. update() should be used if session does not contain persistent object.

6. Which of the following is not a state of object in Hibernate?

- a) Attached()**
- b) Detached()**
- c) Persistent()**
- d) Transient()**

Answer: a

Explanation: Attached() is not a state of object in Hibernate. Detached(), Persistent() and Transient() are the only states in Hibernate.

7. Which of the following is not an inheritance mapping strategies?

- a) Table per hierarchy
- b) Table per concrete class
- c) Table per subclass
- d) Table per class

Answer: d

Explanation: Table per class is not an inheritance mapping strategies.

8. Which of the following is not an advantage of using Hibernate Query Language?

- a) Database independent
- b) Easy to write query
- c) No need to learn SQL
- d) Difficult to implement

Answer: d

Explanation: HQL is easy to implement. Also, to implement it HQL it is not dependent on a database platform.

9. In which file database table configuration is stored?

- a) .dbm
- b) .hbm
- c) .ora
- d) .sql

Answer: b

Explanation: Database table configuration is stored in .hbm file.

10. Which of the following is not an advantage of Hibernate Criteria API?

- a) Allows to use aggregate functions
- b) Cannot order the result set
- c) Allows to fetch only selected columns of result
- d) Can add conditions while fetching results

Answer: b

Explanation: addOrder() can be used for ordering the results.

1. What does Liskov substitution principle specify?

- a) parent class can be substituted by child class
- b) child class can be substituted by parent class
- c) parent class cannot be substituted by child class
- d) No classes can be replaced by each other

Answer: a

Explanation: Liskov substitution principle states that Objects in a program should be replaceable with instances of their sub types without altering the correctness of that program.

2. What will be the correct option of the following Java code snippet?

```
1. interface ICust
2. {
3. }
4. class RegularCustomer implements ICust
5. {
6. }
```

```
7. class OneTimeCustomer implements ICust
8. {
9. }
```

- a) ICust can be replaced with RegularCustomer
- b) RegularCustomer can be replaced with OneTimeCustomer
- c) OneTimeCustomer can be replaced with RegularCustomer
- d) We can instantiate objects of ICust

Answer: a

Explanation: According to Liskov substitution principle we can replace ICust with RegularCustomer or OneTimeCustomer without affecting functionality.

3. What will be the output of the following Java code snippet?

```
1. public class Shape
2. {
3.     public int area()
4.     {
5.         return 1;
6.     }
7. }
8. public class Square extends Shape
9. {
10.    public int area()
11.    {
12.        return 2;
13.    }
14. }
15. class Main()
16. {
17.    public static void main(String[] args)
18.    {
19.        Shape shape = new Shape();
20.        Square square = new Square();
21.        shape = square;
22.        System.out.println(shape.area());
23.    }
24. }
```

a) **Compilation failure**

b) **Runtime failure**

c) **1**

d) **2**

Answer: d

Explanation: Child object can be assigned to parent variable without change in behaviour.

4. What will be the output of the following Java code snippet?

```
1. public class Shape
2. {
3.     public int area()
4.     {
5.         return 1;
6.     }
7. }
8. public class Rectangle extends Shape
9. {
10.    public int area()
11.    {
12.        return 3;
13.    }
14. }
15. class Main()
16. {
17.    public static void main(String[] args)
18.    {
19.        Shape shape = new Shape();
20.        Rectangle rect = new Rectangle();
21.        shape = rect;
22.        System.out.println(shape.area());
23.    }
24. }
```

a) **Compilation failure**

b) **3**

c) **1**

d) **2**

Answer: b

Explanation: Child object can be assigned to parent variable without change in behaviour.

5. What will be the output of the following Java code?

```
1. public class Shape
2. {
3.     public int area()
4.     {
5.         return 1;
6.     }
7. }
8. public class Square extends Shape
9. {
10.    public int area()
11.    {
12.        return 2;
13.    }
14. }
15. class Main()
16. {
17.    public static void main(String[] args)
18.    {
19.        Shape shape = new Shape();
20.        Square square = new Square();
21.        square = shape;
22.        System.out.println(square.area());
23.    }
24. }
```

a) Compilation failure

b) 3

c) 1

d) 2

Answer: a

Explanation: Parent object cannot be assigned to child class.

6. What will be the output of the following Java code?

```
1. public class Shape
2. {
3.     public int area()
```

```
4.         {
5.             return 1;
6.         }
7. }
8. public class Square extends Shape
9. {
10.     public int area()
11.     {
12.         return 2;
13.     }
14. }
15. public class Rectangle extends Shape
16. {
17.     public int area()
18.     {
19.         return 3;
20.     }
21. }
22. class Main()
23. {
24.     public static void main(String[] args)
25.     {
26.         Shape shape = new Shape();
27.         Square square = new Square();
28.         Rectangle rect = new Rectangle();
29.         rect = (Rectangle)shape;
30.         System.out.println(square.area());
31.     }
32. }
```

- a) Compilation failure
- b) 3
- c) Runtime Exception
- d) 2

Answer: c

Explanation: ClassCastException is thrown as we cannot assign parent object to child variable.

7. What will be the output of the following Java code?

```
1. public class Shape
2. {
3.     public int area()
4.     {
5.         return 1;
6.     }
7. }
8. public class Square extends Shape
9. {
10.    public int area()
11.    {
12.        return 2;
13.    }
14. }
15. public class Rectangle extends Shape
16. {
17.    public int area()
18.    {
19.        return 3;
20.    }
21. }
22. class Main()
23. {
24.    public static void main(String[] args)
25.    {
26.        Shape shape = new Shape();
27.        Square square = new Square();
28.        Rectangle rect = new Rectangle();
29.        rect = (Rectangle)square;
30.        System.out.println(square.area());
31.    }
32. }
```

a) Compilation failure

b) 3

c) Runtime Exception

d) 2

Answer: a

Explanation: We cannot assign one child class object to another child class variable.

```
1. interface Shape
2. {
3.     public int area();
4. }
5. public class Square implements Shape
6. {
7.     public int area()
8.     {
9.         return 2;
10.    }
11. }
12. public class Rectangle implements Shape
13. {
14.     public int area()
15.     {
16.         return 3;
17.    }
18. }
```

8. What will be the output of the following Java code?

```
1. public class Shape
2. {
3.     public int area()
4.     {
5.         return 1;
6.     }
7. }
8. public class Square extends Shape
9. {
10.     public int area()
11.     {
```

```
12.         return 2;
13.     }
14. }
15. public class Rectangle extends Shape
16. {
17.     public int area()
18.     {
19.         return 3;
20.     }
21. }
22. class Main()
23. {
24.     public static void main(String[] args)
25.     {
26.         Shape shape = new Shape();
27.         Square square = new Square();
28.         Rectangle rect = new Rectangle();
29.         rect = (Rectangle)square;
30.         System.out.println(square.area());
31.     }
32. }
```

a) Compilation failure

b) 3

c) Runtime Exception

d) 2

Answer: a

Explanation: Interface cannot be instantiated. So we cannot create instances of shape.

9. What will be the output of the following Java code?

```
1. public class Shape
2. {
3.     public int area()
4.     {
5.         return 1;
6.     }
7. }
```



```
8. public class Square extends Shape
9. {
10.     public int area()
11.     {
12.         return 2;
13.     }
14. }
15. public class Rectangle extends Shape
16. {
17.     public int area()
18.     {
19.         return 3;
20.     }
21. }
22. public static void main(String[] args)
23. {
24.     Shape shape = new Square();
25.     shape = new Rectangle();
26.     System.out.println(shape.area());
27. }
```

a) Compilation failure

b) 3

c) Runtime Exception

d) 2

Answer: b

Explanation: With parent class variable we can access methods declared in parent class. If the parent class variable is assigned child class object then it accesses the method of child class.

10. What will be the output of the following Java code?

```
1. public class Shape
2. {
3.     public int area()
4.     {
5.         return 1;
6.     }
7. }
8. public class Square extends Shape
```

```

9. {
10.     public int area()
11.     {
12.         return 2;
13.     }
14. }

15. public class Rectangle extends Shape
16. {
17.     public int area()
18.     {
19.         return 3;
20.     }
21. }

22. public static void main(String[] args)
23. {
24.     Shape square = new Square();
25.     Shape rect = new Rectangle();
26.     square = rect;
27.     System.out.println(square.area());
28. }

```

- a) Compilation failure
- b) 3
- c) Runtime Exception
- d) 2

Answer: b

Explanation: The method of the child class object is accessed. When we reassign objects, the methods of the latest assigned object are accessed.

1. What should the return type of method where there is no return value?

- a) Null
- b) Empty collection
- c) Singleton collection
- d) Empty String

Answer: b

Explanation: Returning Empty collection is a good practice. It eliminates chances of unhandled null pointer exceptions.

2. What data structure should be used when number of elements is fixed?

- a) Array
- b) Array list
- c) Vector
- d) Set

Answer: a

Explanation: Array list has variable size. Array is stored in contiguous memory. Hence, reading is faster. Also, array is memory efficient.

3. What causes the program to exit abruptly and hence its usage should be minimalistic?

- a) Try
- b) Finally
- c) Exit
- d) Catch

Answer: c

Explanation: In case of exit, the program exits abruptly hence would never be able to debug the root cause of the issue.

ii)

```
1. public boolean abc(int num)
2. {
3.     return num % 2 == 1;
4. }
```

- a) i
- b) ii
- c) (i) causes compilation error
- d) (ii) causes compilation error

Answer: b

Explanation: Arithmetic and logical operations are much faster than division and multiplication.

5. Which one of the following causes memory leak?

- a) Release database connection when querying is complete
- b) Use Finally block as much as possible
- c) Release instances stored in static tables
- d) Not using Finally block often

Answer: d

Explanation: Finally block is called in successful as well exception scenarios. Hence, all the connections are closed properly which avoids memory leak.

6. Which of the following is a best practice to measure time taken by a process for execution?

- a) System.currentTimeMillis()
- b) System.nanoTime()
- c) System.getCurrentTime()
- d) System.getProcessingTime()

Answer: b

Explanation: System.nanoTime takes around 1/100000 th of a second whereas System.currentTimeMillis takes around 1/1000th of a second.

- a) Option (i)
- b) Option (ii)
- c) Compilation Error
- d) Option (ii) gives incorrect result

Answer: b

Explanation: Null check must be done while dealing with nested structures to avoid null pointer exceptions.

8. Which of the below is true about java class structure?

- a) The class name should start with lowercase
- b) The class should have thousands of lines of code
- c) The class should only contain those attribute and functionality which it should; hence keeping it short
- d) The class attributes and methods should be public

Answer: c

Explanation: Class name should always start with upper case and contain those attribute and functionality which it should (Single Responsibility Principle); hence keeping it short. The attributes should be usually private with get and set methods.

9. Which of the below is false about java coding?

- a) variable names should be short
- b) variable names should be such that they avoid ambiguity
- c) test case method names should be created as english sentences without spaces
- d) class constants should be used when we want to share data between class methods

Answer: a

Explanation: variable names like i, a, abc, etc should be avoided. They should be real world names which avoid ambiguity. Test case name should explain its significance.

10. Which is better in terms of performance for iterating an array?

- a) for(int i=0; i<100; i++)
- b) for(int i=99; i>=0; i--)
- c) for(int i=100; i<0; i++)
- d) for(int i=99; i>0; i++)

Answer: b

Explanation: reverse traversal of array take half number cycles as compared to forward traversal. The other for loops will go in infinite loop.

1. What will be the output of the following Java code?

```
1.      import java.util.*;
2.
3.      public class genericstack <E>
4.      {
5.
6.          Stack <E> stk = new Stack <E>();
7.
8.          public void push(E obj)
9.          {
10.
11.              stk.push(obj);
12.
13.          }
14.
15.          public E pop()
16.          {
17.
18.              E obj = stk.pop();
19.
20.              return obj;
21.
22.          }
23.      }
24.
25.      class Output
```

```
16.      {
17.          public static void main(String args[])
18.      {
19.          genericstack <String> gs = new genericstack<String>();
20.          gs.push("Hello");
21.          System.out.println(gs.pop());
22.      }
23.  }
```

- a) H
- b) Hello
- c) Runtime Error
- d) Compilation Error

Answer: b

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
Hello
```

2. What will be the output of the following Java code?

```
1.      import java.util.*;
2.      public class genericstack <E>
3.      {
4.          Stack <E> stk = new Stack <E>();
5.          public void push(E obj)
6.          {
7.              stk.push(obj);
8.          }
9.          public E pop()
10.         {
11.             E obj = stk.pop();
12.             return obj;
13.         }
14.     }
15.     class Output
16.     {
17.         public static void main(String args[])
```

```
18.      {
19.          genericstack <Integer> gs = new genericstack<Integer>();
20.          gs.push(36);
21.          System.out.println(gs.pop());
22.      }
23.  }
```

a) 0

b) 36

c) Runtime Error

d) Compilation Error

Answer: b

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
36
```

3. What will be the output of the following Java code?

```
1.      import java.util.*;
2.
3.      public class genericstack <E>
4.      {
5.          Stack <E> stk = new Stack <E>();
6.
7.          public void push(E obj)
8.          {
9.              stk.push(obj);
10.         }
11.
12.         public E pop()
13.         {
14.             E obj = stk.pop();
15.             return obj;
16.         }
17.     }
18.
19.     class Output
20.     {
21.         public static void main(String args[])
22.         {
23.             genericstack <String> gs = new genericstack<String>();
```

```
20.         gs.push("Hello");
21.
22.         System.out.print(gs.pop() + " ");
23.
24.         genericstack <Integer> gs = new genericstack<Integer>();
25.
26.         gs.push(36);
27.
28.         System.out.println(gs.pop());
29.
30.     }
31.
32. }
```

- a) Error
- b) Hello
- c) 36
- d) Hello 36

Answer: d

Explanation: None.

Output:

```
$ javac Output.javac
$ java Output
Hello 36
```

4. What will be the output of the following Java program?

```
1.     import java.util.*;
2.
3.     public class genericstack <E>
4.     {
5.
6.         Stack <E> stk = new Stack <E>();
7.
8.         public void push(E obj)
9.         {
10.
11.             stk.push(obj);
12.
13.         }
14.
15.         public E pop()
16.         {
17.
18.             E obj = stk.pop();
19.
20.             return obj;
21.
22.         }
23.     }
24.
25.     class Output
26.     {
27.
28.         public static void main(String args[])
29.         {
```

```
19.         genericstack <Integer> gs = new genericstack<Integer>();
20.
21.         gs.push(36);
22.
23.         System.out.println(gs.pop());
24.     }
25. }
```

- a) H
- b) Hello
- c) Runtime Error
- d) Compilation Error

Answer: d

Explanation: genericstack's object gs is defined to contain a string parameter but we are sending an integer parameter, which results in compilation error.

Output:

```
$ javac Output.java
$ java Output
```

5. What will be the output of the following Java program?

```
1.     import java.util.*;
2.
3.     public class genericstack <E>
4.     {
5.         Stack <E> stk = new Stack <E>();
6.
7.         public void push(E obj)
8.         {
9.             stk.push(obj);
10.        }
11.
12.        public E pop()
13.        {
14.            E obj = stk.pop();
15.            return obj;
16.        }
17.    }
18.
19.    class Output
20.    {
21.        public static void main(String args[])
22.        {
23.            genericstack <Integer> gs = new genericstack<Integer>();
24.
25.            gs.push(36);
```



```
21.         System.out.println(gs.pop());
```

```
22.     }
```

```
23. }
```

- a) H
- b) Hello
- c) Runtime Error
- d) Compilation Error

Answer: d

Explanation: generic stack object gs is defined to contain a string parameter but we are sending an integer parameter, which results in compilation error.

Output:

```
$ javac Output.java
$ java Output
```

6. Which of these Exception handlers cannot be type parameterized?

- a) catch
- b) throw
- c) throws
- d) all of the mentioned

Answer: d

Explanation: we cannot Create, Catch, or Throw Objects of Parameterized Types as generic class cannot extend the Throwable class directly or indirectly.

7. Which of the following cannot be Type parameterized?

- a) Overloaded Methods
- b) Generic methods
- c) Class methods
- d) Overriding methods

Answer: a

Explanation: Cannot Overload a Method Where the Formal Parameter Types of Each Overload Erase to the Same Raw Type.

1. What are generic methods?

- a) Generic methods are the methods defined in a generic class
- b) Generic methods are the methods that extend generic class methods
- c) Generic methods are methods that introduce their own type parameters
- d) Generic methods are methods that take void parameters

Answer: c

Explanation: Generic methods are methods that introduce their own type parameters. This is similar to declaring a generic type, but the type parameter scope is limited to the method where it is declared. Static and non-static generic methods are allowed, as well as generic class constructors.

2. Which of these type parameters is used for a generic methods to return and accept any type of object?

- a) K
- b) N
- c) T
- d) V

Answer: c

Explanation: T is used for type, A type variable can be any non-primitive type you specify: any class type, any interface type, any array type, or even another type variable.

3. Which of these type parameters is used for a generic methods to return and accept a number?

- a) K
- b) N
- c) T
- d) V

Answer: b

Explanation: N is used for Number.

4. Which of these is an correct way of defining generic method?

- a) <T1, T2, ..., Tn> name(T1, T2, ..., Tn) { /* ... */ }
- b) public <T1, T2, ..., Tn> name<T1, T2, ..., Tn> { /* ... */ }
- c) class <T1, T2, ..., Tn> name[T1, T2, ..., Tn] { /* ... */ }
- d) <T1, T2, ..., Tn> name{T1, T2, ..., Tn} { /* ... */ }

Answer: b

Explanation: The syntax for a generic method includes a type parameter, inside angle brackets, and appears before the method's return type. For static generic methods, the type parameter section must appear before the method's return type.

5. Which of the following allows us to call generic methods as a normal method?

- a) Type Interface
- b) Interface
- c) Inner class
- d) All of the mentioned

Answer: a

Explanation: Type inference, allows you to invoke a generic method as an ordinary method, without specifying a type between angle brackets.

6. What will be the output of the following Java program?

```
1.      import java.util.*;
2.
3.      public class genericstack <E>
4.      {
5.
6.          Stack <E> stk = new Stack <E> ();
7.
8.          public void push(E obj)
9.          {
10.
11.              stk.push(obj);
12.
13.          }
14.
15.          public E pop()
16.          {
17.
18.              E obj = stk.pop();
19.
20.              return obj;
21.          }
22.      }
23.
24.      class Output
```

```
16.      {
17.          public static void main(String args[])
18.      {
19.          genericstack <String> gs = new genericstack<String>();
20.          gs.push("Hello");
21.          System.out.println(gs.pop());
22.      }
23.  }
```

- a) H
- b) Hello
- c) Runtime Error
- d) Compilation Error

Answer: b

Explanation: None.

Output:

```
$ javac Output.javac
$ java Output
Hello
```

7. What will be the output of the following Java program?

```
1.      import java.util.*;
2.      public class genericstack <E>
3.      {
4.          Stack <E> stk = new Stack <E>();
5.          public void push(E obj)
6.          {
7.              stk.push(obj);
8.          }
9.          public E pop()
10.         {
11.             E obj = stk.pop();
12.             return obj;
13.         }
14.     }
15.     class Output
16.     {
17.         public static void main(String args[])
```

```
18.         {
19.             genericstack <Integer> gs = new genericstack<Integer>();
20.             gs.push(36);
21.             System.out.println(gs.pop());
22.         }
23.     }
```

- a) 0
- b) 36
- c) Runtime Error
- d) Compilation Error

Answer: b

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
36
```

8. What will be the output of the following Java program?

```
1.     import java.util.*;
2.
3.     public class genericstack <E>
4.     {
5.         Stack <E> stk = new Stack <E>();
6.
7.         public void push(E obj)
8.         {
9.             stk.push(obj);
10.        }
11.
12.        public E pop()
13.        {
14.            E obj = stk.pop();
15.            return obj;
16.        }
17.    }
18.
19.    class Output
20.    {
21.        public static void main(String args[])
22.        {
23.            genericstack <String> gs = new genericstack<String>();
```

```

20.         gs.push("Hello");
21.         System.out.print(gs.pop() + " ");
22.         genericstack <Integer> gs = new genericstack<Integer>();
23.         gs.push(36);
24.         System.out.println(gs.pop());
25.     }
26. }

```

- a) Error
- b) Hello
- c) 36
- d) Hello 36

Answer: d

Explanation: None.

Output:

```

$ javac Output.javac
$ java Output
Hello 36

```

1. Which of these types cannot be used to initiate a generic type?

- a) Integer class
- b) Float class
- c) Primitive Types
- d) Collections

Answer: c

Explanation: None.

2. Which of these instance cannot be created?

- a) Integer instance
- b) Generic class instance
- c) Generic type instance
- d) Collection instances

Answer: c

Explanation: It is not possible to create generic type instances. Example – “E obj = new E()” will give a compilation error.

3. Which of these data type cannot be type parameterized?

- a) Array
- b) List
- c) Map
- d) Set

Answer: a

Explanation: None.

4. What will be the output of the following Java program?

```

1.     public class BoxDemo
2.     {

```

```

3.         public static <U> void addBox(U u,
4.             java.util.List<Box<U>> boxes)
5.         {
6.             Box<U> box = new Box<>();
7.             box.set(u);
8.             boxes.add(box);
9.         }
10.        public static <U> void outputBoxes(java.util.List<Box<U>> boxes)
11.        {
12.            int counter = 0;
13.            for (Box<U> box: boxes)
14.            {
15.                U boxContents = box.get();
16.                System.out.println("Box #" + counter + " contains [" + boxContents.toString()
17.                counter++;
18.            }
19.        }
20.        public static void main(String[] args)
21.        {
22.            java.util.ArrayList<Box<Integer>> listOfIntegerBoxes = new java.util.ArrayList<>()
23.            BoxDemo.<Integer>addBox(Integer.valueOf(10), listOfIntegerBoxes);
24.            BoxDemo.outputBoxes(listOfIntegerBoxes);
25.        }
26.    }

```

- a) 10
- b) Box #0 [10]
- c) Box contains [10]
- d) Box #0 contains [10]

Answer: d

Explanation: None.

Output:

```

$ javac Output.java
$ java Output
Box #0 contains [10]

```

5. What will be the output of the following Java program?

```

1.     import java.util.*;

```

```

2.     public class genericstack <E>
3.     {
4.         Stack <E> stk = new Stack <E>();
5.         public void push(E obj)
6.         {
7.             stk.push(obj);
8.         }
9.         public E pop()
10.        {
11.            E obj = stk.pop();
12.            return obj;
13.        }
14.    }
15.    class Output
16.    {
17.        public static void main(String args[])
18.        {
19.            genericstack <String> gs = new genericstack<String>();
20.            gs.push("Hello");
21.            System.out.print(gs.pop() + " ");
22.            genericstack <Integer> gs = new genericstack<Integer>();
23.            gs.push(36);
24.            System.out.println(gs.pop());
25.        }
26.    }

```

- a) Error
- b) Hello
- c) 36
- d) Hello 36

Answer: d

Explanation: None.

Output:

```

$ javac Output.java
$ java Output
Hello 36

```

6. What will be the output of the following Java program?

```

1.     import java.util.*;
2.
3.     class Output
4.     {
5.
6.         public static double sumOfList(List<? extends Number> list)
7.         {
8.             double s = 0.0;
9.
10.            for (Number n : list)
11.
12.                s += n.doubleValue();
13.
14.            return s;
15.        }
16.
17.        public static void main(String args[])
18.        {
19.
20.            List<Double> ld = Arrays.asList(1.2, 2.3, 3.5);
21.
22.            System.out.println(sumOfList(ld));
23.
24.        }
25.    }

```

- a) 5.0
- b) 7.0
- c) 8.0
- d) 6.0

Answer: b

Explanation: None.

Output:

```

$ javac Output.java
$ java Output
7.0

```

7. What will be the output of the following Java program?

```

1.     import java.util.*;
2.
3.     class Output
4.     {
5.
6.         public static void addNumbers(List<? super Integer> list)
7.         {
8.
9.             for (int i = 1; i <= 10; i++)
10.
11.                 {
12.
13.                     list.add(i);
14.
15.                 }
16.
17.         }
18.
19.     }

```



```

10.         }
11.         public static void main(String args[])
12.         {
13.             List<Double> ld = Arrays.asList();
14.             addnumbers(10.4);
15.             System.out.println("getList(2)");
16.         }
17.     }

```

- a) 1
- b) 2
- c) 3
- d) 6

Answer: a

Explanation: None.

Output:

```

$ javac Output.java
$ java Output
1

```

8. What will be the output of the following Java program?

```

1.     import java.util.*;
2.     public class genericstack <E>
3.     {
4.         Stack <E> stk = new Stack <E>();
5.         public void push(E obj)
6.         {
7.             stk.push(obj);
8.         }
9.         public E pop()
10.        {
11.            E obj = stk.pop();
12.            return obj;
13.        }
14.    }
15.    class Output
16.    {
17.        public static void main(String args[])

```

```

18.         {
19.             genericstack <Integer> gs = new genericstack<Integer>();
20.             gs.push(36);
21.             System.out.println(gs.pop());
22.         }
23.     }

```

- a) H
- b) Hello
- c) Runtime Error
- d) Compilation Error

Answer: d

Explanation: generic stack object gs is defined to contain a string parameter but we are sending an integer parameter, which results in compilation error.

Output:

```
$ javac Output.java
```

1. Which of these is wildcard symbol?

- a) ?
- b) !
- c) %
- d) &

Answer: a

Explanation: In generic code, the question mark (?), called the wildcard, represents an unknown type.

2. What is use of wildcards?

- a) It is used in cases when type being operated upon is not known
- b) It is used to make code more readable
- c) It is used to access members of super class
- d) It is used for type argument of generic method

Answer: a

Explanation: The wildcard can be used in a variety of situations: as the type of a parameter, field, or local variable; sometimes as a return type (though it is better programming practice to be more specific). The wildcard is never used as a type argument for a generic method invocation, a generic class instance creation, or a supertype.

3. Which of these keywords is used to upper bound a wildcard?

- a) stop
- b) bound
- c) extends
- d) implements

Answer: c

Explanation: None.

4. Which of these is an correct way making a list that is upper bounded by class Number?

- a) List<? extends Number>
- b) List<extends ? Number>
- c) List(? extends Number)
- d) List(? UpperBounds Number)

Answer: a

Explanation: None.

5. Which of the following keywords are used for lower bounding a wild card?

- a) extends**
- b) super**
- c) class**
- d) lower**

Answer: b

Explanation: A lower bounded wildcard is expressed using the wildcard character ('?'), following by the super keyword, followed by its lower bound: .

6. What will be the output of the following Java program?

```
1.    import java.util.*;
2.
3.    class Output
4.    {
5.        public static double sumOfList(List<? extends Number> list)
6.        {
7.            double s = 0.0;
8.            for (Number n : list)
9.                s += n.doubleValue();
10.           return s;
11.        }
12.        public static void main(String args[])
13.        {
14.            List<Integer> li = Arrays.asList(1, 2, 3);
15.            System.out.println(sumOfList(li));
16.        }
```

- a) 0**
- b) 4**
- c) 5.0**
- d) 6.0**

Answer: d

Explanation: None.

Output:

```
$ javac Output.java
$ java Output
6.0
```

7. What will be the output of the following Java program?

```
1.    import java.util.*;
```

```

2.     class Output
3.     {
4.         public static double sumOfList(List<? extends Number> list)
5.         {
6.             double s = 0.0;
7.             for (Number n : list)
8.                 s += n.doubleValue();
9.             return s;
10.        }
11.        public static void main(String args[])
12.        {
13.            List<Double> ld = Arrays.asList(1.2, 2.3, 3.5);
14.            System.out.println(sumOfList(ld));
15.        }
16.    }

```

- a) 5.0
- b) 7.0
- c) 8.0
- d) 6.0

Answer: b

Explanation: None.

Output:

```

$ javac Output.java
$ java Output
7.0

```

8. What will be the output of the following Java program?

```

1.     import java.util.*;
2.     public class genericstack <E>
3.     {
4.         Stack <E> stk = new Stack <E> ();
5.         public void push(E obj)
6.         {
7.             stk.push(obj);
8.         }
9.         public E pop()
10.        {

```

```

11.         E obj = stk.pop();
12.         return obj;
13.     }
14. }
15. class Output
16. {
17.     public static void main(String args[])
18.     {
19.         genericstack <Integer> gs = new genericstack<Integer>();
20.         gs.push(36);
21.         System.out.println(gs.pop());
22.     }
23. }

```

- a) H
- b) Hello
- c) Runtime Error
- d) Compilation Error

Answer: d

Explanation: generic stack object gs is defined to contain a string parameter but we are sending an integer parameter, which results in compilation error.

Output:

```

$ javac Output.javac
$ java Output

```

- 1. Which of the following is not an Enterprise Beans type?**
- a) Doubleton
 - b) Singleton
 - c) Stateful
 - d) Stateless

Answer: a

Explanation: Stateful, Stateless and Singleton are session beans.

- 2. Which of the following is not true about Java beans?**
- a) Implements java.io.Serializable interface
 - b) Extends java.io.Serializable class
 - c) Provides no argument constructor
 - d) Provides setter and getter methods for its properties

Answer: b

Explanation: java.io.Serializable is not a class. Instead it is an interface. Hence it cannot be extended.

- 3. Which file separator should be used by MANIFEST file?**
- a) /
 - b) \
 - c) –
 - d) //

Answer: a

Explanation: MANIFEST file uses classes using / file separator.

4. Which of the following is correct error when loading JAR file with duplicate name?

- a) java.io.NullPointerException**
- b) java.lang.ClassNotFound**
- c) java.lang.ClassFormatError**
- d) java.lang DuplicateClassError**

Answer: c

Explanation: java.lang.ClassFormatError: Duplicate Name error is thrown when .class file in the JAR contains a class whose class name is different from the expected name.

Answer: b

Explanation: JavaBeans do not add any security features to the Java platform.

6. Which of the following is not a feature of Beans?

- a) Introspection**
- b) Events**
- c) Persistence**
- d) Serialization**

Answer: d

Explanation: Serialization is not the feature of Java Beans. Introspection, Customization, Events, Properties and Persistence are the features.

7. What is the attribute of java bean to specify scope of bean to have single instance per Spring IOC?

- a) prototype**
- b) singleton**
- c) request**
- d) session**

Answer: b

Explanation: Singleton scope of bean specifies only one instance per spring IOC container. This is the default scope.

8. Which attribute is used to specify initialization method?

- a) init**
- b) init-method**
- c) initialization**
- d) initialization-method**

Answer: b

Explanation: init-method is used to specify the initialization method.

```
<bean id = "helloWorld" class = "com.bean.HelloWorld" init-method = "init" />
```

9. Which attribute is used to specify destroy method?

- a) destroy**
- b) destroy-method**
- c) destruction**
- d) destruction-method**

Answer: b

Explanation: destroy-method is used to specify the destruction method.

```
<bean id = "helloWorld" class = "com.tutorialspoint.HelloWorld" destroy-method = "destroy" />
```

10. How to specify autowiring by name?

- a) @Qualifier**
- b) @Type**
- c) @Constructor**

d) @Name

Answer: a

Explanation: Different beans of the same class are identified by name.

- | | |
|----|-------------------------------------|
| 1. | <code>@Qualifier("student1")</code> |
| 2. | <code>@Autowired</code> |
| 3. | <code>Student student1;</code> |

1. Which of the following contains both date and time?

- a) java.io.date**
- b) java.sql.date**
- c) java.util.date**
- d) java.util.dateTime**

Answer: d

Explanation: java.util.date contains both date and time. Whereas, java.sql.date contains only date.

2. Which of the following is advantage of using JDBC connection pool?

- a) Slow performance**
- b) Using more memory**
- c) Using less memory**
- d) Better performance**

Answer: d

Explanation: Since the JDBC connection takes time to establish. Creating connection at the application start-up and reusing at the time of requirement, helps performance of the application.

3. Which of the following is advantage of using PreparedStatement in Java?

- a) Slow performance**
- b) Encourages SQL injection**
- c) Prevents SQL injection**
- d) More memory usage**

Answer: c

Explanation: PreparedStatement in Java improves performance and also prevents from SQL injection.

4. Which one of the following contains date information?

- a) java.sql.TimeStamp**
- b) java.sql.Time**
- c) java.io.Time**
- d) java.io.TimeStamp**

Answer: a

Explanation: java.sql.Time contains only time. Whereas, java.sql.TimeStamp contains both time and date.

5. What does setAutoCommit(false) do?

- a) commits transaction after each query**
- b) explicitly commits transaction**
- c) does not commit transaction automatically after each query**
- d) never commits transaction**

Answer: c

Explanation: setAutoCommit(false) does not commit transaction automatically after each query. That saves a lot of time of the execution and hence improves performance.

6. Which of the following is used to call stored procedure?

- a) Statement
- b) PreparedStatement
- c) CallableStatement
- d) CallableStatement

Answer: c

Explanation: CallableStatement is used in JDBC to call stored procedure from Java program.

7. Which of the following is used to limit the number of rows returned?

- a) setMaxRows(int i)
- b) setMinRows(int i)
- c) getMaxrows(int i)
- d) getMinRows(int i)

Answer: a

Explanation: setMaxRows(int i) method is used to limit the number of rows that the database returns from the query.

8. Which of the following is method of JDBC batch process?

- a) setBatch()
- b) deleteBatch()
- c) removeBatch()
- d) addBatch()

Answer: d

Explanation: addBatch() is a method of JDBC batch process. It is faster in processing than executing one statement at a time.

9. Which of the following is used to rollback a JDBC transaction?

- a) rollback()
- b) rollforward()
- c) deleteTransaction()
- d) RemoveTransaction()

Answer: a

Explanation: rollback() method is used to rollback the transaction. It will rollback all the changes made by the transaction.

10. Which of the following is not a JDBC connection isolation levels?

- a) TRANSACTION_NONE
- b) TRANSACTION_READ_COMMITTED
- c) TRANSACTION_REPEATABLE_READ
- d) TRANSACTION_NONREPEATABLE_READ

Answer: d

Explanation: TRANSACTION_NONREPEATABLE_READ is not a JDBC connection isolation level.

1. Which of the below is not a valid design pattern?

- a) Singleton
- b) Factory
- c) Command
- d) Java

Answer: d

Explanation: Design pattern is a general repeatable solution to a commonly occurring problem in software design. There are various patterns available for use in day to day coding problems.

2. Which of the below author is not a part of GOF (Gang of Four)?

- a) Erich Gamma

- b) Gang Pattern**
- c) Richard Helm**
- d) Ralph Johnson**

Answer: b

Explanation: Four authors named Richard Helm, Erich Gamma, Ralph Johnson and John Vlissides published a book on design patterns. This book initiated the concept of Design Pattern in Software development. They are known as Gang of Four (GOF).

3. Which of the below is not a valid classification of design pattern?

- a) Creational patterns**
- b) Structural patterns**
- c) Behavioural patterns**
- d) Java patterns**

Answer: d

Explanation: Java patterns is not a valid classification of design patterns. The correct one is J2EE patterns.

4. Which design pattern provides a single class which provides simplified methods required by client and delegates call to those methods?

- a) Adapter pattern**
- b) Builder pattern**
- c) Facade pattern**
- d) Prototype pattern**

Answer: c

Explanation: Facade pattern hides the complexities of the system and provides an interface to the client using which client can access the system.

5. Which design pattern ensures that only one object of particular class gets created?

- a) Singleton pattern**
- b) Filter pattern**
- c) State pattern**
- d) Bridge pattern**

Answer: a

Explanation: Singleton pattern involves a single class which is responsible to create an object while making sure that only one object gets created. This class provides a way to access the only object which can be accessed directly without need to instantiate another object of the same class.

6. Which design pattern suggests multiple classes through which request is passed and multiple but only relevant classes carry out operations on the request?

- a) Singleton pattern**
- b) Chain of responsibility pattern**
- c) State pattern**
- d) Bridge pattern**

Answer: b

Explanation: Chain of responsibility pattern creates a chain of receiver objects for a particular request. The sender and receiver of a request are decoupled based on the type of request. This pattern is one of the behavioral patterns.

7. Which design pattern represents a way to access all the objects in a collection?

- a) Iterator pattern**
- b) Facade pattern**
- c) Builder pattern**
- d) Bridge pattern**

Answer: a

Explanation: Iterator pattern represents a way to access the elements of a collection object in sequential manner without the need to know its underlying representation.

8. What does MVC pattern stands for?

- a) Mock View Control**
- b) Model view Controller**
- c) Mock View Class**
- d) Model View Class**

Answer: b

Explanation: Model represents an object or JAVA POJO carrying data. View represents the visualization of the data that model contains. The controller acts on both model and view. It is usually used in web development.

Answer: a
Explanation: Design pattern is a logical concept. Various classes and frameworks are provided to enable users to implement these design patterns.

10. Which design pattern works on data and action taken based on data provided?

- a) Command pattern**
- b) Singleton pattern**
- c) MVC pattern**
- d) Facade pattern**

Answer: a

Explanation: Command pattern is a data driven design pattern. It is a behavioral pattern. A request is wrapped under an object as command and passed to the invoker object. The invoker object looks for the appropriate object which can handle this command and passes this command to the corresponding object which executes the command.

1. Which mode allows us to run program interactively while watching source code and variables during execution?

- a) safe mode**
- b) debug mode**
- c) successfully run mode**
- d) exception mode**

Answer: b

Explanation: Debug mode allows us to run program interactively while watching source code and variables during execution.

2. How can we move from one desired step to another step?

- a) breakpoints**
- b) System.out.println**
- c) logger.log**
- d) logger.error**

Answer: a

Explanation: Breakpoints are inserted in code. We can move from one point to another in the execution of a program.

3. Which part stores the program arguments and startup parameters?

- a) debug configuration**
- b) run configuration**
- c) launch configuration**
- d) project configuration**

Answer: c

Explanation: Launch configuration stores the startup class, program arguments and vm arguments.

4. How to deep dive into the execution of a method from a method call?

- a) F3**
- b) F5**
- c) F7**

d) F8

Answer: b

Explanation: F5 executes currently selected line and goes to the next line in the program. If the selected line is a method call, debugger steps into the associated code.

5. Which key helps to step out of the caller of currently executed method?

- a) F3**
- b) F5**
- c) F7**
- d) F8**

Answer: c

Explanation: F7 steps out to the caller of the currently executed method. This finishes the execution of the current method and returns to the caller of this method.

6. Which view allows us to delete and deactivate breakpoints and watchpoints?

- a) breakpoint view**
- b) variable view**
- c) debug view**
- d) logger view**

Answer: a

Explanation: The Breakpoints view allows us to delete and deactivate breakpoints and watchpoints. We can also modify their properties.

7. What is debugging an application which runs on another java virtual machine on another machine?

- a) virtual debugging**
- b) remote debugging**
- c) machine debugging**
- d) compiling debugging**

Answer: b

Explanation: Remote debugging allows us to debug applications which run on another Java virtual machine or even on another machine. We need to set certain flags while starting the application.

```
java -Xdebug -Xnoagent \
-Djava.compiler=NONE \
-Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=5005.
```

8. What happens when the value of variable change?

- a) changed value pop on the screen**
- b) variable changes are printed in logs**
- c) dump of variable changes are printed on the screen on end of execution**
- d) variable tab shows variables highlighted when values change**

Answer: d

Explanation: When a variable value changes, the value in variable tab is highlighted yellow in eclipse.

9. Which perspective is used to run a program in debug view?

- a) java perspective**
- b) eclipse perspective**
- c) debug perspective**
- d) jdbc perspective**

Answer: c

Explanation: We can switch from one perspective to another. Debug perspective shows us the breakpoints, variables, etc.

10. How does eclipse provide the capability for debugging browser actions?

- a) internal web browser
- b) chrome web browser
- c) firefox web browser
- d) internet explorer browser

Answer: a

Explanation: Eclipse provides internal web browser to debug browser actions.

1. Servlet are used to program which component in a web application?

- a) client
- b) server
- c) tomcat
- d) applet

Answer: b

Explanation: A servlet class extends the capabilities of servers that host applications which are accessed by way of a request-response programming model.

2. Which component can be used for sending messages from one application to another?

- a) server
- b) client
- c) mq
- d) webapp

Answer: c

Explanation: Messaging is a method of communication between software components or applications. MQ can be used for passing message from sender to receiver.

3. How are java web applications packaged?

- a) jar
- b) war
- c) zip
- d) both jar and war

Answer: d

Explanation: war are deployed on apache servers or tomcat servers. With Spring boot and few other technologies tomcat is brought on the machine by deploying jar.

4. How can we connect to database in a web application?

- a) oracle sql developer
- b) toad
- c) JDBC template
- d) mysql

Answer: c

Explanation: JDBC template can be used to connect to database and fire queries against it.

5. How can we take input text from user in HTML page?

- a) input tag
- b) inoutBufferedReader tag
- c) meta tag
- d) scanner tag

Answer: a

Explanation: HTML provides various user input options like input, radio, text, etc.

6. Which of the below is not a javascript framework for UI?

- a) Vaadin

- b) AngularJS**
- c) KendoUI**
- d) Springcore**

Answer: d

Explanation: Springcore is not a javascript framework. It is a comprehensive programming and configuration model for enterprise applications based on java.

7. Which of the below can be used to debug front end of a web application?

- a) Junit**
- b) Fitnesse**
- c) Firebug**
- d) Mockito**

Answer: c

Explanation: Firebug integrates with firefox and enables to edit, debug and monitor CSS, HTML and javascript of any web page.

8. What type of protocol is HTTP?

- a) stateless**
- b) stateful**
- c) transfer protocol**
- d) information protocol**

Answer: a

Explanation: HTTP is a stateless protocol. It works on request and response mechanism and each request is an independent transaction.

9. What does MIME stand for?

- a) Multipurpose Internet Messaging Extension**
- b) Multipurpose Internet Mail Extension**
- c) Multipurpose Internet Media Extension**
- d) Multipurpose Internet Mass Extension**

Answer: b

Explanation: MIME is an acronym for Multi-purpose Internet Mail Extensions. It is used for classifying file types over the Internet. It contains type/subtype e.g. application/msword.

10. What is the storage capacity of single cookie?

- a) 2048 MB**
- b) 2048 bytes**
- c) 4095 bytes**
- d) 4095 MB**

Answer: c

Explanation: Storage capacity of cookies is 4095 bytes/cookie.

1. How does applet and servlet communicate?

- a) HTTP**
- b) HTTPS**
- c) FTP**
- d) HTTP Tunneling**

Answer: d

Explanation: Applet and Servlet communicate through HTTP Tunneling.

Answer: a
Explanation: A new process is started with each client request and that corresponds to initiate a heavy OS level process for each client request.

3. Which class provides system independent server side implementation?

- a) Socket**
- b) ServerSocket**
- c) Server**
- d) ServerReader**

Answer: b

Explanation: ServerSocket is a java.net class which provides system independent implementation of server side socket connection.

4. What happens if ServerSocket is not able to listen on the specified port?

- a) The system exits gracefully with appropriate message**
- b) The system will wait till port is free**
- c) IOException is thrown when opening the socket**
- d) PortOccupiedException is thrown**

Answer: c

Explanation: public ServerSocket() creates an unbound server socket. It throws IOException if specified port is busy when opening the socket.

5. What does bind() method of ServerSocket offer?

- a) binds the serversocket to a specific address (IP Address and port)**
- b) binds the server and client browser**
- c) binds the server socket to the JVM**
- d) binds the port to the JVM**

Answer: a

Explanation: bind() binds the server socket to a specific address (IP Address and port). If address is null, the system will pick an ephemeral port and valid local address to bind socket.

6. Which of the below are common network protocols?

- a) TCP**
- b) UDP**
- c) TCP and UDP**
- d) CNP**

Answer: c

Explanation: Transmission Control Protocol(TCP) and User Datagram Protocol(UDP) are the two common network protocol. TCP/IP allows reliable communication between two applications. UDP is connection less protocol.

7. Which class represents an Internet Protocol address?

- a) InetAddress**
- b) Address**
- c) IP Address**
- d) TCP Address**

Answer: a

Explanation: InetAddress represents an Internet Protocol address. It provides static methods like getByAddress(), getByName() and other instance methods like getHostName(), getHostAddress(), getLocalHost().

8. What does local IP address start with?

- a) 10.X.X.X**
- b) 172.X.X.X**
- c) 192.168.X.X**
- d) 10.X.X.X, 172.X.X.X, or 192.168.X.X**

Answer: d

Explanation: Local IP addresses look like 10.X.X.X, 172.X.X.X, or 192.168.X.X.

9. What happens if IP Address of host cannot be determined?

- a) The system exit with no message**
- b) UnknownHostException is thrown**
- c) IOException is thrown**
- d) Temporary IP Address is assigned**

Answer: b

Explanation: UnknownHostException is thrown when IP Address of host cannot be determined. It is an extension of IOException.

10. What is the java method for ping?

- a) hostReachable()**
- b) ping()**
- c) isReachable()**
- d) portBusy()**

Answer: c

Explanation: inet.isReachable(5000) is a way to ping a server in java.

1. How constructor can be used for a servlet?

- a) Initialization**
- b) Constructor function**
- c) Initialization and Constructor function**
- d) Setup() method**

Answer: c

Explanation: We cannot declare constructors for interface in Java. This means we cannot enforce this requirement to any class which implements Servlet interface.

Also, Servlet requires ServletConfig object for initialization which is created by container.

Answer: b
Explanation: ServletConfig object is created after the constructor is called and before init() is called. So, servlet init parameters cannot be accessed in the constructor.

Answer: b
Explanation: Servlets execute on Server and doesn't have GUI. Applets execute on browser and has GUI.

4. Which of the following code is used to get an attribute in a HTTP Session object in servlets?

- a) session.getAttribute(String name)**
- b) session.alterAttribute(String name)**
- c) session.updateAttribute(String name)**
- d) session.setAttribute(String name)**

Answer: a

Explanation: session has various methods for use.

5. Which method is used to get three-letter abbreviation for locale's country in servlets?

- a) Request.getISO3Country()**
- b) Locale.getISO3Country()**
- c) Response.getISO3Country()**
- d) Local.retrieveISO3Country()**

Answer: a

Explanation: Each country is usually denoted by a 3 digit code. ISO3 is the 3 digit country code.

6. Which of the following code retrieves the body of the request as binary data?

- a) DataInputStream data = new InputStream()**
- b) DataInputStream data = response.getInputStream()**
- c) DataInputStream data = request.getInputStream()**
- d) DataInputStream data = request.fetchInputStream()**

Answer: c

Explanation: InputStream is an abstract class. getInputStream() retrieves the request in binary data.

7. When destroy() method of a filter is called?

- a) The destroy() method is called only once at the end of the life cycle of a filter**
- b) The destroy() method is called after the filter has executed doFilter method**
- c) The destroy() method is called only once at the beginning of the life cycle of a filter**
- d) The destroyer() method is called after the filter has executed**

Answer: a

Explanation: destroy() is an end of life cycle method so it is called at the end of life cycle.

8. Which of the following is true about servlets?

- a) Servlets execute within the address space of web server**
- b) Servlets are platform-independent because they are written in java**
- c) Servlets can use the full functionality of the Java class libraries**
- d) Servlets execute within the address space of web server, platform independent and uses the functionality of java class libraries**

Answer: d

Explanation: Servlets execute within the address space of a web server. Since it is written in java it is platform independent. The full functionality is available through libraries.

9. How is the dynamic interception of requests and responses to transform the information done?

- a) servlet container**
- b) servlet config**
- c) servlet context**
- d) servlet filter**

Answer: d

Explanation: Servlet has various components like container, config, context, filter. Servlet filter provides the dynamic interception of requests and responses to transform the information.

Answer: b

Explanation: URL rewriting, using session object, using cookies, using hidden fields are session tracking techniques.

1. Which of the following is used for session migration?

- a) Persisting the session in database**
- b) URL rewriting**
- c) Create new database connection**
- d) Kill session from multiple sessions**

Answer: a

Explanation: Session migration is done by persisting session in database. It can also be done by storing session in memory on multiple servers.

2. Which of the below is not a session tracking method?

- a) URL rewriting**
- b) History**
- c) Cookies**
- d) SSL sessions**

Answer: b

Explanation: History is not a session tracking type. Cookies, URL rewriting, Hidden form fields and SSL sessions are session tracking methods.

3. Which of the following is stored at client side?

- a) URL rewriting**
- b) Hidden form fields**
- c) SSL sessions**
- d) Cookies**

Answer: d

Explanation: Cookies are stored at client side. Hence, it is advantageous in some cases where clients disable cookies.

4. Which of the following leads to high network traffic?

- a) URL rewriting**
- b) Hidden form fields**
- c) SSL sessions**
- d) Cookies**

Answer: a

Explanation: URL rewriting requires large data transfer to and from the server which leads to network traffic and access may be slow.

5. Which of the following is not true about session?

- a) All users connect to the same session**
- b) All users have same session variable**
- c) Default timeout value for session variable is 20 minutes**
- d) New session cannot be created for a new user**

Answer: c

Explanation: Default timeout value for session variable is 20 minutes. This can be changed as per requirement.

Answer: a
Explanation: SessionIDs are stored in cookies, URLs and hidden form fields.

7. What is the maximum size of cookie?

- a) 4 KB**
- b) 4 MB**
- c) 4 bytes**
- d) 40 KB**

Answer: a

Explanation: The 4K is the maximum size for the entire cookie, including name, value, expiry date etc. To support most browsers, it is suggested to keep the name under 4000 bytes, and the overall cookie size under 4093 bytes.

8. How can we invalidate a session?

- a) session.discontinue()**
- b) session.invalidate()**
- c) session.disconnect()**
- d) session.falsify()**

Answer: b

Explanation: We can invalidate session by calling session.invalidate() to destroy the session.

9. Which method creates unique fields in the HTML which are not shown to the user?

- a) User authentication**
- b) URL writing**
- c) HTML Hidden field**
- d) HTML invisible field**

Answer: c

Explanation: HTML Hidden field is the simplest way to pass information but it is not secure and a session can be hacked easily.

10. Which object is used by spring for authentication?

- a) ContextHolder**
- b) SecurityHolder**
- c) AnonymousHolder**
- d) SecurityContextHolder**

Answer: d
Explanation: The SessionManagementFilter checks the contents of the SecurityContextRepository against the current contents of the SecurityContextHolder to determine whether user has been authenticated during the current request by a non-interactive authentication mechanism, like pre authentication or remember me.

1. Which page directive should be used in JSP to generate a PDF page?

- a) contentType**
- b) generatePdf**
- c) typePDF**
- d) contentPDF**

Answer: a

Explanation: <%page contentType="application/pdf"> tag is used in JSP to generate PDF.

2. Which tag should be used to pass information from JSP to included JSP?

- a) Using <%jsp:page> tag**
- b) Using <%jsp:param> tag**
- c) Using <%jsp:import> tag**
- d) Using <%jsp:useBean> tag**

Answer: b

Explanation: <%jsp:param> tag is used to pass information from JSP to included JSP.

3. Application is instance of which class?

- a) javax.servlet.Application**
- b) javax.servlet.HttpContext**
- c) javax.servlet.Context**
- d) javax.servlet.ServletContext**

Answer: d

Explanation: Application object is wrapper around the ServletContext object and it is an instance of a javax.servlet.ServletContext object.

Answer: a
Explanation: _jspService() method is created by JSP container. Hence, it should not be overridden.

5. Which option is true about session scope?

- a) Objects are accessible only from the page in which they are created**
- b) Objects are accessible only from the pages which are in same session**
- c) Objects are accessible only from the pages which are processing the same request**
- d) Objects are accessible only from the pages which reside in same application**

Answer: b

Explanation: Object data is available till session is alive.

Answer: a
Explanation: Default value "true" depicts automatic buffer flushing.

7. Which one is the correct order of phases in JSP life cycle?

- a) Initialization, Cleanup, Compilation, Execution**
- b) Initialization, Compilation, Cleanup, Execution**
- c) Compilation, Initialization, Execution, Cleanup**
- d) Cleanup, Compilation, Initialization, Execution**

Answer: c

Explanation: The correct order is Compilation, Initialization, Execution, Cleanup.

8. "request" is instance of which one of the following classes?

- a) Request**
- b) HttpRequest**
- c) HttpServletRequest**
- d) ServletRequest**

Answer: c

Explanation: request is object of HttpServletRequest.

9. Which is not a directive?

- a) include**
- b) page**
- c) export**
- d) useBean**

Answer: c

Explanation: Export is not a directive.

10. Which is mandatory in <jsp:useBean /> tag?

- a) id, class**
- b) id, type**
- c) type, property**
- d) type, id**

Answer: a

Explanation: The useBean searches existing object and if not found creates an object using class.

1. Which one of the following is correct for directive in JSP?

- a) <%@directive%>**
- b) <%!directive%>**
- c) <%directive%>**
- d) <%=directive%>**

Answer: a

Explanation: Directive is declared as <%@directive%>.

2. Which of the following action variable is used to include a file in JSP?

- a) jsp:setProperty**
- b) jsp:getProperty**
- c) jsp:include**
- d) jsp:plugin**

Answer: c

Explanation: jsp:include action variable is used to include a file in JSP.

3. Which attribute uniquely identification element?

- a) ID**
- b) Class**
- c) Name**
- d) Scope**

Answer: a

Explanation: ID attribute is used to uniquely identify action element.

4. “out” is implicit object of which class?

- a) javax.servlet.jsp.PrintWriter**
- b) javax.servlet.jsp.SessionWriter**
- c) javax.servlet.jsp.SessionPrinter**
- d) javax.servlet.jsp.JspWriter**

Answer: d

Explanation: JspWriter object is referenced by the implicit variable out which is initialized automatically using methods in the PageContext object.

5. Which object stores references to the request and response objects?

- a) sessionContext
- b) pageContext
- c) HttpSession
- d) sessionAttribute

Answer: b

Explanation: pageContext object contains information about directives issued to JSP page.

6. What temporarily redirects response to the browser?

- a) <jsp:forward>
- b) <%@directive%>
- c) response.sendRedirect(URL)
- d) response.setRedirect(URL)

Answer: c

Explanation: response.sendRedirect(URL) directs response to the browser and creates a new request.

7. Which tag is used to set a value of a JavaBean?

- a) <c:set>
- b) <c:param>
- c) <c:choose>
- d) <c:forward>

Answer: a

Explanation: <c:set> is used to set a value of a java.util.Map object.

Answer: b
Explanation: <!--comment--> is an HTML comment. <%--comment--%> is JSP comment.

9. Java code is embedded under which tag in JSP?

- a) Declaration
- b) Scriptlet
- c) Expression
- d) Comment

Answer: b

Explanation: Scriptlet is used to embed java code in JSP.

10. Which of the following is not a directive in JSP?

- a) page directive
- b) include directive
- c) taglib directive
- d) command directive

Answer: d

Explanation: command directive is not a directive in JSP.

1. What are the components of a marker interface?

- a) Fields and methods
- b) No fields, only methods
- c) Fields, no methods
- d) No fields, No methods

Answer: d

Explanation: Marker interface in Java is an empty interface in Java.

2. Which of the following is not a marker interface?

- a) Serializable
- b) Cloneable

- c) Remote
- d) Reader

Answer: d

Explanation: Reader is not a marker interface. Serializable, Cloneable and Remote interfaces are marker interface.

3. What is not the advantage of Reflection?

- a) Examine a class's field and method at runtime
- b) Construct an object for a class at runtime
- c) Examine a class's field at compile time
- d) Examine an object's class at runtime

Answer: c

Explanation: Reflection inspects classes, interfaces, fields and methods at a runtime.

4. How private method can be called using reflection?

- a) getDeclaredFields
- b) getDeclaredMethods
- c) getMethods
- d) getFields

Answer: b

Explanation: getDeclaredMethods gives instance of java.lang.reflect.Method.

5. How private field can be called using reflection?

- a) getDeclaredFields
- b) getDeclaredMethods
- c) getMethods
- d) getFields

Answer: a

Explanation: getDeclaredFields gives instance of java.lang.reflect.Fields.

6. What is used to get class name in reflection?

- a) getClass().getName()
- b) getClass().getFields()
- c) getClass().getDeclaredFields()
- d) new getClass()

Answer: a

Explanation: getClass().getName() is used to get a class name from object in reflection.

7. How method can be invoked on unknown object?

- a) obj.getClass().getDeclaredMethod()
- b) obj.getClass().getDeclaredField()
- c) obj.getClass().getMethod()
- d) obj.getClass().getObject()

Answer: c

Explanation: obj.getClass().getMethod is used to invoke a method on unknown object obj.

8. How to get the class object of associated class using Reflection?

- a) Class.forName("className")
- b) Class.name("className")
- c) className.getClass()
- d) className.getClassName()

Answer: a

Explanation: forName(String className) returns the Class object associated with the class or interface with the given string name.

- 9. What does Class.forName(“myreflection.Foo”).newInstance() return?**
- a) An array of Foo objects**
 - b) class object of Foo**
 - c) Calls the newInstance() method of Foo class**
 - d) Foo object**

Answer: d
Explanation: Class.forName(“myreflection.Foo”) returns the class object of Foo and newInstance() would return a new object.

- 10. What does foo.getClass().getMethod(“doSomething”, null) return?**
- a) doSomething method instance**
 - b) Method is returned and we can call the method as method.invoke(foo,null);**
 - c) Class object**
 - d) Exception is thrown**

Answer: b
Explanation: foo.getClass().getMethod() returns a method and we can call the method using method.invoke();

- 1. Autocloseable was introduced in which Java version?**
- a) java SE 7**
 - b) java SE 8**
 - c) java SE 6**
 - d) java SE 4**

Answer: a
Explanation: Java 7 introduced autocloseable interface.

- 2. What is the alternative of using finally to close resource?**
- a) catch block**
 - b) autocloseable interface to be implemented**
 - c) try block**
 - d) throw Exception**

Answer: b
Explanation: Autocloseable interface provides close() method to close this resource and any other underlying resources.

- 3. Which of the below is a child interface of Autocloseable?**
- a) Closeable**
 - b) Close**
 - c) Auto**
 - d) Cloneable**

Answer: a
Explanation: A closeable interface extends autocloseable interface. A Closeable is a source or destination of data that can be closed.

- 4. It is a good practise to not throw which exception in close() method of autocloseable?**
- a) IOException**
 - b) CustomException**
 - c) InterruptedException**
 - d) CloseException**

Answer: c
Explanation: InterruptedException interacts with a thread’s interrupted status and runtime misbehavior is likely to occur

if an *InterruptedException* is suppressed.

5. What will be the output of the following Java code snippet?

```
1. try (InputStream is = ...)
2. {
3.     // do stuff with is...
4. }
5. catch (IOException e)
6. {
7.     // handle exception
8. }
```

- a) Runtime Error
- b) IOException
- c) Compilation Error
- d) Runs successfully

Answer: d

Explanation: Using java 7 and above, AutoCloseable objects can be opened in the try-block (within the ()) and will be automatically closed instead of using the finally block.

6. What is the difference between AutoCloseable and Closeable?

- a) Closeable is an interface and AutoCloseable is a concrete class
- b) Closeable throws IOException; AutoCloseable throws Exception
- c) Closeable is a concept; AutoCloseable is an implementation
- d) Closeable throws Exception; AutoCloseable throws IOException

Answer: b

Explanation: Closeable extends AutoCloseable and both are interfaces. Closeable throws IOException and AutoCloseable throws Exception.

7. What is the use of Flushable interface?

- a) Flushes this stream by writing any buffered output to the underlying stream
- b) Flushes this stream and starts reading again
- c) Flushes this connection and closes it
- d) Flushes this stream and throws FlushException

Answer: a

Explanation: Flushable interface provides flush() method which Flushes this stream by writing any buffered output to the underlying stream.

8. Which version of java added Flushable interface?

- a) java SE 7
- b) java SE 8
- c) java SE 6
- d) java SE 5

Answer: d

Explanation: Flushable and Closeable interface are added in java SE 5.

Answer: a
Explanation: close() closes the stream but it flushes it first.

10. AutoCloseable and Flushable are part of which package?

- a) Autocloseable java.lang; Flushable java.io
- b) Autocloseable java.io; Flushable java.lang
- c) Autocloseable and Flushable java.io
- d) Autocloseable and Flushable java.lang

Answer: a

Explanation: Autocloseable is a part of java.lang; Flushable is a part of java.io.

1. Which of below is not a dependency management tool?

- a) Ant
- b) Maven
- c) Gradle
- d) Jenkins

Answer: d

Explanation: Jenkins is continuous integration system. Ant, Maven, Gradle is used for build process.

2. Which of the following is not a maven goal?

- a) clean
- b) package
- c) install
- d) debug

Answer: d

Explanation: clean, package, install are maven goals. Debug is used finding and resolving of defects.

3. Which file is used to define dependency in maven?

- a) build.xml
- b) pom.xml
- c) dependency.xml
- d) version.xml

Answer: b

Explanation: pom.xml is used to define dependency which is used to package the jar. POM stands for project object model.

4. Which file is used to specify the packaging cycle?

- a) build.xml
- b) pom.xml
- c) dependency.xml
- d) version.xml

Answer: a

Explanation: Project structure is specified in build.xml.

5. Which environment variable is used to specify the path to maven?

- a) JAVA_HOME
- b) PATH
- c) MAVEN_HOME
- d) CLASSPATH

Answer: c

Explanation: MAVEN_HOME should be set to the bin folder of maven installation.

6. Which of the below is a source code management tool?

- a) Jenkins
- b) Maven
- c) Git

d) Hudson

Answer: c

Explanation: Source code management tools help is version control, compare different versions of code, crash management, etc. Git, SVN are popular source code management tools.

Answer: a
Explanation: As a part of jenkins job, we can run junits, fitnessse, test coverage reports, call shell or bat scripts, etc.

8. Which command can be used to check maven version?

- a) mvn -ver**
- b) maven -ver**
- c) maven -version**
- d) mvn -version**

Answer: d

Explanation: mvn -version can be used to check the version of installed maven from command prompt.

9. Which of the following is not true for Ant?

- a) It is a tool box**
- b) It provides lifecycle management**
- c) It is procedural**
- d) It doesn't have formal conventions**

Answer: b

Explanation: Ant doesn't provide lifecycle management. Maven provides lifecycle.

10. Which maven plugin creates the project structure?

- a) dependency**
- b) properties**
- c) archetype**
- d) execution**

Answer: c

Explanation: Archetype is the maven plugin which creates the project structure.

1. Which version of Java introduced annotation?

- a) Java 5**
- b) Java 6**
- c) Java 7**
- d) Java 8**

Answer: a

Explanation: Annotation were introduced with Java 5 version.

2. Annotation type definition looks similar to which of the following?

- a) Method**
- b) Class**
- c) Interface**
- d) Field**

Answer: c

Explanation: Annotation type definition is similar to an interface definition in which the keyword interface is preceded by the sign @.

3. Which of the following is not pre defined annotation in Java?

- a) @Deprecated**
- b) @Overriden**
- c) @SafeVarags**
- d) @FunctionInterface**

Answer: b
Explanation: @Overriden is not a pre defined annotation in Java. @Depricated, @Override, @SuppressWarnings, @SafeVarargs and @FunctionInterface are the pre defined annotations.
Answer: a
Explanation: Annotations which are applied to other annotations are called meta annotations.

5. Which one of the following annotations is not used in Hibernate?

- a) @Entity**
- b) @Column**
- c) @Basic**
- d) @Query**

Answer: d
Explanation: @Query is not an annotation used in Hibernate.

6. Which one of the following is not ID generating strategy using @GeneratedValue annotation?

- a) Auto**
- b) Manual**
- c) Identity**
- d) Sequence**

Answer: b
Explanation: Auto, Table, Identity and Sequence are the ID generating strategies using @GeneratedValue annotation.

7. Which one of the following is not an annotation used by Junit with Junit4?

- a) @Test**
- b) @BeforeClass**
- c) @AfterClass**
- d) @Ignored**

Answer: d
Explanation: @Test, @Before, @BeforeClass, @After, @AfterClass and @Ignores are the annotations used by Junit with Junit4.

8. Using which annotation non visible or private method can be tested?

- a) @VisibleForTesting**
- b) @NonVisibleForTesting**
- c) @Visible**
- d) @NonVisible**

Answer: a
Explanation: Using @VisibleForTesting annotation private or non visible method can be tested.

9. Which of the following annotation is used to avoid execution of Junits?

- a) @NoTest**
- b) @explicit**
- c) @avoid**
- d) @ignore**

Answer: d
Explanation: @ignore annotation is used to avoid execution of Junits.

10. Which is the Parent class of annotation class?

- a) Class**
- b) Object**
- c) Main**
- d) Super**

Answer: b

Explanation: Object is the parent class of annotation class.