**Directions ::** On your answer sheet, mark the letter of the best answer to each question. Each question is worth 2 points for a total of 100 points. Write only on your answer sheet or scratch paper. Put your test ID and name on your answer sheet.

1. **What is output by the code below?**

int ap = 789;

ap = ap % 10;

System.out.print( ap );

a. 7 b. 8 c. 9 d. 0 e. 6

ANS: C

2. **Which of the following reserved words is used to access the class you are in as a complete Object?**

a. this b. super c. that d. superman e. object

ANS: A

3. **What is the output?**

System.out.print(3 + 3 \* 3);

a. 18 b. 12 c. 9 d. 0 e. 10

ANS: B

4. **If you are writing a class that implements the Comparable interface, which method must be present in the implementing class?**

a. add() b. toString() c. equals() d. compareTo() e. comparable()

ANS: D

5. **What is output by the code below?**

System.out.print("\\dog\\cat");

a. dog b. dogcat c. \\dog\\cat d. \dog\cat e. catdog\\\\

ANS: D

6. **Which of the following reserved words is used in a sub class to access a public method of the parent class?**

a. this b. super c. that d. superman e. object

ANS: B

7. **What is returned by the call** getIt(9) **?**

public static int getIt(int num)

{

int ans = 0;

if( num >=2 )

{

if( num >= 7)

ans += 2;

else

ans += 3;

}

ans += 4;

return ans;

}

a. 4 b. 2 c. 6 d. 7 e. 9

ANS: C

8. **What is output by the code below?**

**//client code**

int[] list = {2,12,11,45,52,36,5,3,1};

System.out.println( go(list) );

**//go receives an array and returns a calculated value**

public static double go( int[] ray )

{

int val = 0;

for(int i=0; i < ray.length; i = i + 2)

val = val + ray[i];

return val;

}

a. 45 b. 96 c. 56 d. 65 e. 71

ANS: E

9. **What is output by the code below?**

**//client code**

int[] list = {2,12,11,45,52,36};

System.out.println( go(list) );

**//go receives an array and returns a calculated value**

public static int go( int[] ray )

{

int val = Integer.MIN\_VALUE;

for(int i=0; i < ray.length; i++)

if( ray[i] > val )

val = ray[i];

return val;

}

a. 45 b. 2 c. 12 d. 52 e. 36

ANS: D

10. **What is the code shown below working to locate?**

public static int go( int[] ray )

{

int val = Integer.MIN\_VALUE;

for(int i=0; i < ray.length; i++)

if( ray[i] > val )

val = ray[i];

return val;

}

a. The code is trying to locate all of the even numbers. b. The code is trying to locate the biggest number.

c. The code is trying to locate the smallest number. d. The code is trying to locate all of the odd numbers.

e. The code is trying to locate the biggest even number.

ANS: B

11. **What is output by the code below?**

System.out.println(Math.pow(2,4));

a. 32 b. 16 c. 16.0 d. 32.0 e. 8.0

ANS: C

12. **What is output by the code below?**

System.out.println("rsteva".substring(3,4));

a. r b. s c. t d. e e. v

ANS: D

13. **What is output by the code below?**

public class CS{

public void one(){

System.out.print("one");

}

public void two(){

System.out.print("two");

one();

}

}

**//code in the main of another class**

CS test = new CS();

test.one();

test.two();

test.two();

a. one b. onetwo c. onetwoone d. onetwoonetwo e. onetwoonetwoone

ANS: E

14. **What is output by the code below?**

String a="abcdefsdfg";

String b="ems";

for(int i=0; i < b.length(); i++)

{

out.print(a.indexOf(b.substring(i, i+1)));

out.print(" ");

}

a. 4 -1 6 b. 3 6 c. 3 0 5 d. There is no output due to a run-time error.

e. There is no output due to a syntax-time error.

ANS: A

15. **Which of the following would correctly fill blank <1> ?**

ArrayList<String> aList = **<1>**

a. new String(); b. new Integer(); c. new Object<String>();

d. new ArrayList<String>(); e. new ArrayList<Integer>();

ANS: D

16. **What is returned by the call** check(99)  **?**

public static boolean check( int x )

{

return ( x % 2 == 0 );

}

a. 1 b. 0 c. true d. false e. 10

ANS: D

17. **What is the value of x after this code segment is executed?**

String s = "computers";

String t = s.substring(3,7);

int x = s.indexOf(t + s.substring(7, 8));

a. 4 b. -1 c. 5 d. 3 e. 2

ANS: D

18. **What is output by the code below?**

ArrayList<String> bList = new ArrayList();

bList.add(0,"one");

bList.add("two");

bList.set(0,"three");

bList.add(1,"four");

bList.set(1,"five");

System.out.println(bList.get(0).substring(0,1));

a. o b. t c. f d. n e. h

ANS: B

19. **What output by the code below?**

ArrayList<Integer> cList = new ArrayList<Integer>();

cList.add(0,1);

cList.add(2);

cList.set(0,3);

cList.add(1, 4);

cList.set(1, 5);

int i = cList.get(0);

System.out.println( i );

a. 1 b. 2 c. 3 d. 4 e. 5

ANS: C

20. **What is the output?**

public class It

{

private int x = 78;

public It( ) {

x = 99;

}

public String toString(){

return "" + x;

}

}

**//code in main in some class**

It anIt = new It();

System.out.println(anIt);

a. 0 b. 1 c. 99 d. 78 e. 50

ANS: C

21. **What is output by the code below?**

public class It

{

private int x = 78;

public It( ) {

x = 99;

}

public It( int y ) {

x = y;

}

public String toString(){

return "" + x;

}

}

**//code in main in some class**

ArrayList<It> itList = new ArrayList<It>();

itList.add(new It());

itList.add(new It(8));

System.out.println(itList);

a. [99 , 8 ] b. [78 , 8 ] c. [99, 78 ] d. [8 , 8 ] e. [8,78]

ANS: A

22. **What is output by the code below?**

ArrayList<Integer> dList = new ArrayList<Integer>();

dList.add(8);

dList.add(2);

dList.add(0,5);

dList.add(0,6);

dList.add(9);

System.out.println(dList);

a. [8, 2, 5, 6, 9] b. [6, 5, 2, 8, 9] c. [6, 5, 8, 2, 9] d. [2, 8, 5, 6, 9]

e. [8, 2, 5, 9, 6]

ANS: C

23. **What is output by the code below?**

public class Yo{

private int myX = 0;

public Yo( ) { myX = 99; }

public String toString(){

return "" + myX;

}

}

public class At extends Yo{

private int myZ = 0;

public At( ) {

super();

myZ = 100;

}

public String toString(){

return myZ + " " + super.toString();

}

}

**//code in main in some class**

At anAt = new At();

System.out.println(anAt);

a. 0 99 b. 0 100 c. 100 99 d. 0 0 e. 99 99

ANS: C

24. **When** println() **is displaying an Object, which one of the Object’s methods does** println() **call?**

a. add() b. toString() c. equals() d. compareTo() e. String()

ANS: B

25. **Which class is the parent class of all other Objects in the Java language?**

a. String b. Integer c. Byte d. This e. Object

ANS: E

26. **What is output by the code below?**

double wNum = 5 / 2;

wNum = wNum \* 5.0;

System.out.printf(wNum);

a. 12.50 b. 10.00 c. 12.00 d. 5.00 e. 13.00

ANS: B

27. **What type of sort is funSearch() ?**

public static int funSearch( int[] list, int val )

{

int len = list.length;

int small = 0, large = len-1, middle=0;

while(small<large+1)

{

middle =(small + large)/2;

if (list[middle] == val)

return middle;

else

if (list[middle]>val)

large = middle-1;

else

small = middle+1;

}

return -1;

}

a. linearSearch b. selectionSearch c. binarySearch d. quickSearch

e. insertionSearch

ANS: C

28. **What is output by the code below?**

int[][] mat = {{2,3,4,5},

{6,7,8,9,10}};

System.out.println(mat[1].length);

a. 2 b. 4 c. 5 d. 0 e. 1

ANS: C

29. **Which of the following could fill the blanks shown below?**

**class X \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Y \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ A, B, C**

a. extends implements b. extends extends c. implements extends d. implements implements

e. A and B

ANS: A

30. **What value does dog(9) return ?**

public static int dog(int x){

if(x<1)

return 1;

else

return x-dog(x-3);

}

a. 1 b. 2 c. 3 d. 4 e. 5

ANS: E

31. **What type of sort is funSort() ?**

public static void funSort( int[] ray )

{

for(int i=0; i<ray.length-1; i++)

{

int big = i;

for(int x=i+1; x<ray.length; x++)

{

if(ray[x]>ray[big])

big = x;

}

if(big!=i){

int temp = ray[big];

ray[big] = ray[i];

ray[i] = temp;

}

}

}

a. bubbleSort b. insertionSort c. selectionSort d. quickSort e. mergeSort

ANS: C

32. **What is the output of line 2?**

public class X{

public void one(){

System.out.print("Xone");

}

public void two(){

System.out.print("Xtwo");

}

}

public class Y extends X{

public void one(){

System.out.print("Yone");

}

public void two(){

System.out.print("Ytwo");

}

public void testIt(){

one();

super.one();

two();

super.two();

}

}

**//code in the main of another class**

Y code = new Y();

code.one();

code.testIt();  **//line 2**

a. XoneXtwoYoneYtwo b. XoneXoneYoneYone c. XoneYoneXtwoYtwo d. YoneXoneYtwoXtwo

e. YoneXtwoYtwoXtwo

ANS: D