Note: Correct responses are based on Java, J2sdk v 6.0, from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (i. e. `error` is an answer choice) and any necessary Java 2 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

QUESTION 1

What is $A0_{16}$ plus 110001_2 ?

A. 11010001_2 B. 207_{10} C. 11001001_2 D. 131_{16} E. $D3_{16}$

QUESTION 2

What is output by the code to the right?

A. 24 B. 23

C. 19 D. 20

E. 21

```
double m = 4.75;
m = m++ * 4;
System.out.println((int)m);
```

QUESTION 3

What is a possible output by the code to the right?

A. 1.125 B. -2

C. -3.75 D. -12

E. 2

```
Integer x = (int)(Math.random()*12);
x-=10;
System.out.println(x);
```

QUESTION 4

What is output by the code to the right?

A. -1 B. 7654321

C. 1 D. 0

E. There is no output due to an infinite loop.

```
int x = 7;
while(x > 0)
  x--;
System.out.println(x);
```

QUESTION 5

What is output by the code to the right?

A. olitan B. transmetro7

C. nsmetro D. opolita

E. There is no output due to a runtime error.

```
String str;
str="Transmetropolitan";
str=str.substring(str.indexOf('o'),7);
System.out.println(str);
```

QUESTION 6

What is output by the code to the right?

A. 20 B. 5 C. 0

D. 19 E. -1

```
int[] x = new int[10];
for(int i=1; i<100; i++)
   x[i%5]++;
out.println(x[5]);
```

QUESTION 7

You are writing a program that has 2 boolean variables, `a` and `b`. In order for a method to be called, `a` and `b` must have opposite values. Which of the following lines of code give the best solution?

A. `if(a == false && b == true)`

B. `if(a || b)`

C. `if(a == b)`

D. `if(a == true || b == false)`

E. `if(a != b)`

QUESTION 8

What is output by the code to the right, given the value of **<*1>** ?

	<*1>	Output
A.	28	absent
B.	32	tardy
C.	29	work
D.	18	tardy
E.	44	work

```
int time = <*1>;
String x;
switch(time/10){
   case 0: x = "start class"; break;
   case 1:
   case 2: x = "tardy"; break;
   case 3:
   case 4: x = "work";
   default: x = "absent";
}
out.println(x);
```

QUESTION 9

A board game awards a victory point for every 3 coins a person has at the end of the game. Which of the following lines of code yields the best solution?

A. `victoryPoints %= 3;`

B. `victoryPoints = coins/3%3;`

C. `victoryPoints = coins/3;`

D. `victoryPoints /= 3;`

E. `victoryPoints %= 3;`

QUESTION 10

Which method best replaces **<*1>** in the code to the right such that a team is considered defeated when its core is depleted?

A.
```
public boolean isDefeated()
{
   return core <= 0;
}
```

B.
```
public int isDefeated()
{
   return core;
}
```

C.
```
public boolean isDefeated()
{
   return getCore();
}
```

D.
```
public int isDefeated(int core)
{
   return core;
}
```

E.
```
public boolean isDefeated(int core)
{
   return core <= 0;
}
```

```
public class Team
{
  private int core, health;

  public Team()
  {
     core = 5;
     health = 10;
  }

  public int getCore()
  {  return core; }

  public int getHealth()
  {  return health; }

  <*1>
}
```

QUESTION 11

Which of the following finds the cube root of a value?

A. `Math.cbrt(x)`

B. `Math.pow(x/3)`

C. `Math.cuberoot(x)`

D. `Math.cube(x)`

E. `Math.exp(x,0.333)`

QUESTION 12

What is output by the code to the right?

A. `$ae.00`

B. `$30.00`

C. `$30x`

D. `30x`

E. `$1e`

```
out.printf("$%x",30);
```

QUESTION 13

What is output by the code to the right?

A. `Dr Who`

B. `Dr`
`Who`

C. `Dr\nWho`

D. `Dr Who`

E. There is no output due to runtime error

```
out.print("Dr\nWho");
```

QUESTION 14

What is the output by the code to the right?

A. `94`

B. `128`

C. `196`

D. `61`

E. `77`

```
int sum = 0;
int[][] list = {{19,7,42,20},
   {38,6,46,23},{40,4,37,33}};

for(int i=0; i<list.length; i++)
  for(int j=0; j<list[i].length; j++)
     if(list[i][j]%2==0)
        sum+=list[i][j];
     else
        sum-=list[i][j];

out.println(sum);
```

QUESTION 15

What is the output by the code to the right?

A. `-50`

B. `-150`

C. `-40`

D. `-132`

E. `100`

```
int x = 100;
for(int i = -40; i<-61; i--)
   x = i - x;
out.println(x);
```

QUESTION 16

What is the output by the code to the right?

A. `chlucylielcharliey`

B. `charlielucy`

C. `chylielxy`

D. `chlucylielchlucyliey`

E. There is no output due to an infinite loop.

```
String x = "charlie";
String y = "lucy";
String z = x.replaceAll("ar",y) +
           y.replaceAll("uc", x);
out.println(z);
```

QUESTION 17

What is the output by the code to the right?

A. `63`

B. `51`

C. `0`

D. `80`

E. `10`

```
out.println(80 >> 3 & 63);
```

QUESTION 18

What is equivalent to the boolean expression to the right?

A. `A || (B && C)`

B. `A && (B || C)`

C. `!(A || B && C || A)`

D. `true`

E. `false`

```
A && B || C && A
```

QUESTION 19

What is the output by the code to the right?

A. `[2, 3, 3, 9]`

B. `[3, 3, 3, 9]`

C. `[9, 3, 3, 1]`

D. `[5, 9, 3, 1]`

E. There is no output due to a runtime error.

```
ArrayList<Integer> list;
list = new ArrayList<Integer>();
list.add(2);
list.add(9);
list.set(0,5);
list.add(3);
list.remove(0);
list.remove(1);
list.add(3);
list.add(2,3);
list.add(1);
out.println(list);
```

QUESTION 20

What is output by the code to the right?

A. `01102`

B. `ABBAC`

C. `4`

D. `SMITH`

E. There is no output due to a runtime error.

```
String x = "SMITH";
String y = "";
for(int i=0; i<x.length(); i++)
  y+=x.charAt(i)%(i+1);
out.println(y);
```

QUESTION 21

What is output by the code to the right?

A. `AMBO KINGMBO KINBO KIO K`

B. `O K`

C. `MAMBO KINGS`

D. `SGNIK OBMAM`

E. There is an index out of bounds exception.

```
String n = "MAMBO KINGS";
while (!n.isEmpty())
{
  n = n.substring(1,n.length()-1);
  out.print(n);
}
```

QUESTION 22

What is returned by the method call `outcome(6,3);`

A. `FLYOUT 3`

B. `HIT`

C. `FLYOUT 6`

D. `GRAND SLAM`

E. `OUT`

QUESTION 23

Which method call will result in a `GRAND SLAM`?

A. `outcome(12,5);`

B. `outcome(16,4);`

C. `outcome(9,19);`

D. `outcome(20,4);`

E. `outcome(19,5);`

```
public String outcome(int x, int y)
{
  if (x >= 1 && x <= 9)
     return "FLYOUT "+x;
  else if(x <= 18)
     if(y == 1 && x < 15)
        return "DOUBLE PLAY";
     else if(y == 3)
        return "SACRIFICE";
     else
        return "OUT";
  else if(y == 2)
     return "RBI";
  else if(y > 4)
     return "GRAND SLAM";
  else
     return "HIT";
}
```

QUESTION 24

Which of the following correctly instantiates an object based on the code to the right?

I. `Discard pile = new Discard();`

II. `Discard pile2 = new Expedition("RED");`

III. `Expedition pile3 = new Discard();`

IV. `Expedition pile4 = new Expedition("RED");`

A. I and IV
B. I, III, and IV
C. IV only
D. I, II, III, and IV
E. I, II, and IV

QUESTION 25

Which of the following correctly replaces **<*1>** in the code to the right such that an integer is placed into the ArrayList if the String is the same as the instance String and the Integer is greater than the last value in the ArrayList?

A
```
if (!color.equals(c))
    return false;
else
   add(x);
return true;
```

B.
```
if (color.equals(c) && x > getValue())
    add(x);
```

C.
```
if (color.equals(c))
   if (x <= pile.get(pile.size()-1))
       return false;
pile.add(x);
return true;
```

D.
```
if (x > getValue())
    add(x);
else if (color.equals(c))
   add(x);
return true;
```

E.
```
if (!color.equals(c) || x <= getValue())
    return false;
add(x);
return true;
```

QUESTION 26

What does the `getValue` method in `Expedition` alter in the code to the right?

A. it returns the middle of the pile.

B. it removes x values from the back of the pile and returns the new last value.

C. it rotates the values from the pile to the front of the pile x times.

D. it removes x values from the front of the pile and returns the back value.

E. it alters nothing

```
public class Discard
{
  private ArrayList<Integer> pile;

  public Discard()
  {
     pile = new ArrayList<Integer>();
  }

  public Integer getValue()
  {
     return pile.get(pile.size()-1);
  }

  public void add(Integer x)
  {
     pile.add(x);
  }

  public Integer removeValue()
  {
     return pile.remove(pile.size()-1);
  }
}

public class Expedition extends Discard
{
  private String color;

  public Expedition(String color)
  {
     super();
     this.color = color;
  }

  public boolean add(String c,
                     Integer x)
  {
     <*1>
  }

  public Integer getValue(int x)
  {
     for(int i=1; i<x; i++)
        removeValue();
     return super.getValue();
  }
}
```

QUESTION 27

What does the list look like at the end of the code to the right?

A. `[32, 10, 1, 38, 20, 42, 0]`

B. `[0, 42, 20, 38, 1, 32, 10]`

C. `[0, 1, 10, 20, 32, 38, 42]`

D. `[0, 0, 0, 0, 0, 0, 0]`

E. `[32, 10, 38, 1, 20, 0, 42]`

QUESTION 28

What type of data structure does is being created in the code at right?

A. priority queue

B. max heap

C. binary search tree

D. min heap

E. circular linked list

```
public void add(ArrayList<Integer>
                list, int x)
{
  boolean isFound = false;
  int loc = 0;
  if(list.isEmpty())
  {
     list.add(x);
     return;
  }
  while(!isFound)
  {
     int left = 2*loc+1;
     int right = 2*loc+2;
     if(x < list.get(loc))
     {
        while(list.size()<=left)
           list.add(0);
        if(list.get(left)==0)
        {
           list.set(left,x);
           isFound = true;
        }
        else
           loc = left;
     }
     else
     {
        while(list.size()<=right)
           list.add(0);
        if(list.get(right)==0)
        {
           list.set(right,x);
           isFound = true;
        }
        else
           loc = right;
     }
  }
}

////////////////////////////////
// client code
ArrayList<Integer> list;
list = new ArrayList<Integer>();
add(list,32);
add(list,10);
add(list,1);
add(list,38);
add(list,20);
add(list,42);
```

QUESTION 29

What is returned by the method call `mystery(4)`?

A. `-1` B. `0` C. `-3` D. `-4` E. `-2`

QUESTION 30

What is returned by the method call `mystery(7)`?

A. `-15` B. `-16` C. `-17` D. `-14` E. `-13`

```
public int mystery(int y)
{
  if (y > 0)
    return mystery(y-1)+mystery(y-2);
  return y;
}
```

QUESTION 31

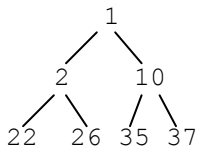
Which of the following is considered the fastest sort?

A. stooge B. insertion C. merge D. bubble E. selection

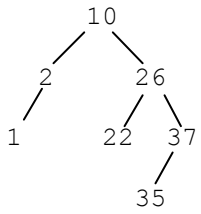
QUESTION 32

How would a min heap tree look for the following numbers: 10, 2, 1, 26, 37, 22, 35?

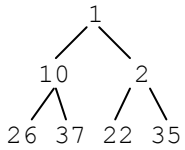
A.



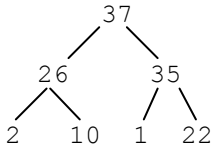
B.



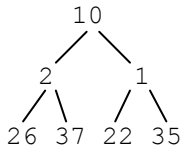
C.



D.



E.



QUESTION 33

What is the output by `//line 1` in the code to the right?

A. `28`

B. `19`

C. `25`

D. `17`

E. `24`

QUESTION 34

What is the output by `//line 2` in the code to the right?

A. `[46, 24, 28, 17, 25, 19]`

B. `[19, 25, 17, 28, 24, 46]`

C. `[17, 19, 24]`

D. `[17, 19]`

E. `[19, 17]`

QUESTION 35

Which of the following code keeps removing the top of the stack until the value on the stack is no longer greater than the value being pushed onto the stack?

A.
```
while(st.push(value) < st.peek())
     st.pop();
```

B.
```
while(!st.isEmpty() && value < st.peek())
     st.pop();
  st.push(value);
```

C.
```
while(!st.isEmpty() && value < st.peek())
     st.push(value);
```

D.
```
while(value < st.peek())
     st.push(value);
```

E.
```
while(value < st.peek())
     st.pop();
  st.push(value);
```

```
Stack<Integer> st;
st = new Stack<Integer>();
st.push(19);
st.push(25);
st.pop();
st.push(17);
st.push(28);
out.println(st.peek());//line 1
st.push(24);
st.push(46);
st.pop();
st.pop();
st.pop();
out.println(st);   //line 2
```

QUESTION 36

What is the output by `//line 1` in the code to the right?

A. `Jimbo`

B. `Vera`

C. `Hans`

D. `Gonzo`

E. `Napoli`

QUESTION 37

What is output by `//line 2` in the code to the right?

A. `Kaiser`

B. `Dirk`

C. `Napoli`

D. `Gonzo`

E. There is no output due to runtime error.

```
LinkedList<String> list;
list = new LinkedList<String>();
list.add("Gonzo");
list.add("Jimbo");
list.add("Hunter");
list.add("Hans");
list.add("Vera");
list.add("Napoli");

Iterator<String> iter;
iter =  list.iterator();
iter.next();
iter.next();
iter.next();
out.println(iter.next()); // line 1

list.add("Dirk");
list.add(2,"Kaiser");
list.remove(1);
list.remove(3);
out.println(iter.next()); // line 2
```

QUESTION 38

What is output by the code to the right?

A. `1028` B. `true` C. `448` D. `11100` E. `56`

```
System.out.println(28 << 4);
```

QUESTION 39

What string `s` would cause the code to the right to return true?

A. `aided`

B. `baracuda`

C. `bobby`

D. `cat dog`

E. `elated`

```
s.matches("a.*d")
```

QUESTION 40

What is output by the code to the right?

A. `yay for-word-chuck--`

B. `-a- for –word-chuck-`

C. `-a- for-`

D. `yay for sword-chucks`

E. `for word chuck`

```
String x = "yay for sword-chucks";
String[] list = x.split("y|s");
for(String st:list)
   out.print(st+"-");
```