1. What does mystery(9870) return?

```java
public int mystery(int n) {
    int m = 0;
    while (n > 0) {
        m = 10*m + n%10;
        n = n/10;
    }
    return m;
}
```

(a) 987  
(b) 0  
(c) 7890  
(d) (*) 789  
(e) 9870

2. What are the values of a and b after the following loop?

```java
int a = 1;
int b = 0;
for (a = 0; b < 10 || a < 20; a = a + 2) {
    a = a + b;
    b = b + a + 1;
}
```

(a) a is 54, and b is 53  
(b) a is 10, and b is 16  
(c) a is 28, and b is 45  
(d) a is 12, and b is 16  
(e) (*) a is 30, and b is 45
3. What are the values of these three expressions, respectively?

\[
\begin{align*}
12 / 8 & \times 3 \\
12 / 8d & \times 3 \\
12 / 8 & \times \text{(double)}3
\end{align*}
\]

(a) 4, 4.5, and 3.0  
(b) 4.5, 4.5, and 4.5  
(c) 3, 3.0, and 3.0  
(d) 3.0, 3.0, and 4.5  
(e) (*) 3, 4.5, and 3.0

4. Which expression evaluates to true whenever \( a \) has the value \( \text{true} \)?  
(All of the expressions are legal Java.)

(a) \( a \ \&\& \ \neg b \)  
(b) \( a \ \&\& \ \neg a \)  
(c) (*) \( a == b || \neg b \)  
(d) \( 0.15 == 0.2 - 0.05 == a \)  
(e) \( a == \text{!!false} \)

5. This declaration:

```java
Java.awt.Color \( w; \)
```

(a) (*) creates a variable \( w \) with initial value `null`.  
(b) creates a variable \( w \) referring to a newly created `java.awt.Color` object, which by default represents the colour black.  
(c) creates a variable \( w \) with initial value `0`.  
(d) creates a variable \( w \) with an unspecified initial value.  
(e) causes a syntax error, as it is not permitted to declare a variable of reference type without creating the associated object.
6. What does mysteryFlag() draw on the screen?

```java
public void mysteryFlag() {
    int z = 300;
    SimpleCanvas sc = new SimpleCanvas("What am I?", z, z);
    sc.setForegroundColour(new java.awt.Color(0,255,0));
    for (int i = 0; i < z; i++) sc.drawLine(0,0,i,z-i);
    sc.setForegroundColour(new java.awt.Color(0,0,255));
    for (int i = 0; i < z; i++) sc.drawLine(i,z-i,z,z);
}
```

(a) A square split horizontally into two coloured rectangles.
(b) A square split vertically into two coloured rectangles.
(c) (*) A square split diagonally into two coloured triangles, one in the top-left half and one in the bottom-right half.
(d) A square split diagonally into two coloured triangles, one in the top-right half and one in the bottom-left half.
(e) Nothing.

7. What is the value of red == xxx after these two statements, and why?

```java
java.awt.Color red = new java.awt.Color(255,0,0);
java.awt.Color xxx = new java.awt.Color(255,0,0);
```

(a) (*) false: they are references to different objects, even though the objects contain identical data.
(b) true: they both represent the colour red.
(c) A reference with the same value as xxx: an assignment expression has the same value as the expression on the right hand side of the assignment.
(d) We cannot tell: it depends on details of the implementation that are hidden from the users.
(e) A runtime error occurs: == can only be used for primitive types.
8. The source code for a class
   (a) must explicitly define exactly one constructor.
   (b) (*) can explicitly define zero, one or more constructors.
   (c) must explicitly define at least one constructor.
   (d) can explicitly define at most one constructor.
   (e) should never contain any explicitly defined constructors.

9. What is the value of a[7] after the following statements?

   ```java
   int[] a;
   a = new int[8];
   a[0] = 1;
   a[1] = 1;
   for (int i = 2; i < 8; i++) a[i] = a[i-1] - a[i-2];
   ```

   (a) -2
   (b) -1
   (c) 0
   (d) (*) 1
   (e) 2

10. The variable int runTime contains the running time of a movie in minutes. You wish to convert it to the normal hours-and-minutes notation (e.g. 133 minutes = 2 hours and 13 minutes). Which piece of code is correct?

   ```java
   (a) int hours = runTime/60;
       int minutes = (runTime - hours*60)/60;
   (b) int hours = runTime%60;
       int minutes = runTime;
   (c) (*)
       int minutes = runTime%60;
       int hours = runTime/60;
   (d) int minutes = runTime%60;
       int hours = runTime-60*minutes;
   (e) int hours = runTime-60;
       int minutes = runTime/60;
   ```
11. A method in the class **BankAccount** (defined as in the lectures) returns `true` if and only if the target object has a strictly higher balance than the argument object (*strictly* means that the method should return `false` if the balances are equal).

Which of these definitions work correctly? (All of them are legal Java.)

```java
// Method 1
public boolean higherBalance(BankAccount other) {
    return balance > other.balance;
}

// Method 2
public boolean higherBalance(BankAccount other) {
    if (getBalance() > other.getBalance()) return this;
    else return other;
}

// Method 3
public boolean higherBalance(BankAccount other) {
    if (balance <= other.getBalance()) return false;
    else return true;
}
```

(a) All three
(b) The 3rd only
(c) The 1st and 2nd only
(d) (*) The 1st and 3rd only
(e) The 2nd and 3rd only
12. Consider this class with a single method.

```java
public class Doubler {
    public void doubleIt(int[] a) {
        a[0] = 2*a[0];
        a = null;
    }
}
```

What happens when the class is compiled, and the following sequence of statements (which is in a method of another class) is compiled and executed?

```java
int[] x = {5,6,7};
Doubler d = new Doubler();
d.doubleIt(x);
System.out.println(x[0]);
```

(a) The code for `Doubler` doesn’t compile, because the class has no constructors.
(b) The code for `Doubler` doesn’t compile, because a parameter variable is read-only and cannot be assigned a value.
(c) The program crashes with a `NullPointerException`.
(d) The value 5 is printed in the terminal window.
(e) (*) The value 10 is printed in the terminal window.
13. Assume BankAccount is defined as in the lectures, and consider the following sequence of statements.

```java
BankAccount b1 = new BankAccount("gfr",1,0);
BankAccount b2 = new BankAccount("rlw",2,0);
BankAccount temp = b1;
b1.deposit(1000);
b2.deposit(2000);
temp.withdraw(500);
temp = b2;
temp.deposit(1000);
```

What are the final balances of the accounts owned by “gfr” and “rlw”, respectively?

(a) 1000 and 2000  
(b) (*) 500 and 3000  
(c) 1000 and 3000  
(d) 500 and 2500  
(e) 1500 and 2000

14. One approximation of $\pi$ uses the equation

$$\frac{1}{4} \pi \sqrt{2} = 1 + \frac{1}{3} - \frac{1}{5} - \frac{1}{7} + \frac{1}{9} + \frac{1}{11} - \ldots$$

Which of these methods correctly calculate the approximation of $\pi$ to a specified number of terms? (All of them are legal Java.)

// Method 1
```java
public double pi(int numTerms) {
    double approx = 0;
    for (int i=0; i < numTerms; i++) {
        double term = 1/(double)(2*i+1);
        if(i%4==0 || i%4==1) approx += term;
        else approx -= term;
    }
    return approx*4/Math.sqrt(2);
}
```
// Method 2
public double pi(int numTerms) {
    double approx = 0;
    int mult = 1;
    for (int i=0; i < numTerms; i=i+2) {
        double term1 = 1d/(2*i+1);
        double term2 = 1d/(2*(i+1)+1);
        approx += mult*(term1 + term2);
        mult = -mult;
    }
    return approx*4/Math.sqrt(2);
}

// Method 3
public double pi(int numTerms) {
    double approx = 0;
    for (int i=0; i < numTerms; i++) {
        double term = 1d/(2*i+1);
        switch(i%4) {
            case 0:
            case 1:
                approx += term; break;
            case 2:
            case 3:
                approx -= term; break;
            default: System.out.println("ERROR");
        }
    }
    return approx*4/Math.sqrt(2);
}

(a) Method 1 only
(b) Method 2 only
(c) Method 3 only
(d) (*) Methods 1 and 3 only
(e) All three
The next six questions refer to the following source code, which defines a class that represents circles.

```java
public class Circle{
    // Represents a circle with its centre at (centreX, centreY)
    // and with the specified radius
    private double centreX;
    private double centreY;
    private double radius;

    public Circle(double x, double y, double r) {
        centreX = x;
        centreY = y;
        radius = r;
    }

    public Circle(double r) {
        this(0,0,r);
    }

    public Circle() {
        this(1d);
    }

    // area() returns the area of the circle
    public double area() {
        return Math.PI * radius * radius;
    }

    // scale() enlarges/shrinks the circle
    public void scale(double factor) {
        radius = radius * factor;
    }

    // bigger() returns the circle with the larger area
    public Circle bigger(Circle other) {
        if (this.area() > other.area()) return this;
        else return other;
    }

    // duplicate() returns a circle identical to this one
    public Circle duplicate() {
        return new Circle(centreX, centreY, radius);
    }
}
```
// the mystery method does something unclear
public boolean mystery(double x, double y) {
    double z = (x-centreX)*(x-centreX);
    z = z + (y-centreY)*(y-centreY);
    return z > radius*radius;
}


15. What are the instance variables of Circle?

(a) x, y, and r
(b) area, scale, bigger, duplicate, and mystery
(c) (*) centreX, centreY, and radius
(d) z
(e) x, y, r, factor, and other

16. Which of these statements correctly construct a Circle?

1. Circle c = Circle(5);
2. Circle c = new Circle(1,2,1);
3. Circle c = new Circle();
4. Circle c = Circle(1,2,1).new();

(a) 1, 2, and 3 only
(b) 2 only
(c) 3 only
(d) 4 only
(e) (*) 2 and 3 only
17. During the following sequence of statements, how many Circle objects are constructed?

```java
Circle c1;
Circle c2;
Circle c3;
Circle c4;
c1 = new Circle(1);
c2 = c1;
c3 = c1.duplicate();
c4 = c3.bigger(c1);
```

(a) 1  
(b) (*) 2  
(c) 3  
(d) 4  
(e) 5

18. What does the mystery method do?

(a) It returns the distance between the circle and the point (x, y).  
(b) (*) It tests if (x, y) is outside the circle.  
(c) It tests if the area of the circle is greater than the length of the line joining (0, 0) to (x, y).  
(d) It returns the radius of the smallest circle containing both (x, y) and (centreX, centreY).  
(e) It tests if (x, y) is exactly on the edge of the circle.
19. What value is stored in `c_area` after the following code (rounded to six decimal places)?

```java
Circle c = new Circle(2);
c.scale(2);
double c_area = c.area();
```

(a) 3.141593
(b) 4.000000
(c) 16.000000
(d) 12.566371
(e) (*) 50.265482

20. If `c1` is a circle with centre (2, 2) and radius 1, and `c2` is a circle with centre (1, 1) and radius 2, what is the value of `c1.bigger(c2)`

(a) (*) An object reference equal to `c2`.
(b) A reference to a new `Circle` object representing a circle with centre (2, 2) and radius 1.
(c) A reference to a new `Circle` object representing a circle with centre (2, 2) and radius 2.
(d) An object reference equal to `c1`.
(e) A reference to a new `Circle` object representing a circle with centre (1, 1) and radius 2.