CS 100: Prelim 2

March 18, 1999
7:30-9:00pm

____
(Print Name)

____
(Signature)

____
(Student ID)

Initial each of the following five(5) pages. For the sake of partial credit, you must show work and comment your solution fragments as appropriate. If you need space, write on the reverse side.

1. __________ (20 pts)

2. __________ (20 pts)

3. __________ (20 pts)

4. __________ (20 pts)

5. __________ (20 pts)

Total = __________ (100 pts)

1. (a) Suppose n is an initialized int variable and that the following fragment is executed. Indicate clearly what the output would be. You must show work to receive credit.

```
int n = 20;
String s = "";  // The empty string.
if (n>=100)
{
   System.out.println(n);
   n = n-100;
   s = s + "C"
}
else if (n>=10)
{
   System.out.println(n);
   n = n-10;
   s = s + "X";
}
else
{
   System.out.println(n)
   n = n-1;
   s = s + "I"

}
System.out.println(n);
s = s + s;
System.out.println(s);
```

(b) Assume that s refers to a String object of length 2 and that B is a boolean variable. Write a fragment that assigns to B the value true if the two characters in the string are the same. The value false should be assigned to B if the two characters in the string are different. Use the equals and substring methods in the class String. (Recall that the latter has two integer parameters that designate the first position and one plus the last position of the substring to be extracted.)

2. Indicate what the output would be if the `main` method in the class `Q2` is executed:

```
public class A
{
   public static int F(int x)
   {
      int y = x+1;
      System.out.println(" y = " + y);
      return y;
   }
   public static int G(int z)
   {
      z = 2*F(z);
      System.out.println(" z = " + z);
      return z;
   }
}

public class Q2
{
   public static void main(String args[])
   {
      int x,y,z;
      x = 1; y = 10; z = 30;
      x = A.F(y);
      System.out.println(" x = " + x + " y = " + y + " z = " + z);

      x = 1; y = 10; z = 30;
      z = A.G(x);
      System.out.println(" x = " + x + " y = " + y + " z = " + z);
   }
}
```

As you step through the execution, take care to distinguish between actual parameters and formal parameters. Do not overlook the print statements “inside” the methods `F` and `G`. Show work if you expect to receive partial credit.

3. Implement the `nearlyPrime` method in the class `A` and the `main` method in the class `B` so that they perform as specified. Your implementation of `main` must make effective use of `nearlyPrime`. For your information, an integer *a* is divisible by an integer *b* if there is a zero remainder when *a* is divided by *b*.

```
public class A
{
  // Yields true if n is divisible by 2 or 3 and is also divisible by 5 or 7.
   public static boolean nearlyPrime(int n)
  {
       < Write your implementation of this below. >
  }
}

public class B
{
   // Prints the sum of all positive integers <= 1000000 that have
   // the property that they are divisible by 2 or 3 and also divisible by 5 or 7.
   public static void main(String args[])
   {
       < Write your implementation of this below. >
   }
}
```

4. Implement the method `isLaterThan` in the class `SimpleDate` and the `main` method in the class `Q4` so that they perform as specified. You may assume that the method `random` is already implemented. To receive full credit, your implementation of `main` must make effective use of the methods in `SimpleDate.`

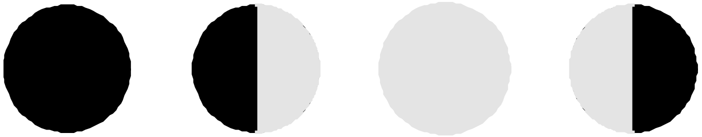
```
// An instance of this class is a date where m is the index of
// the month (1 = January, 2 = February, etc.) and d is the day.
public class SimpleDate
{
   private int m;
   private int d;

   // Constructor creates a date with month index mVal and day dVal.
   public SimpleDate(int mVal, int dVal)
   {
      m = mVal;
      d = dVal;
   }

   // Yields true if this date is later in the year than D
   public boolean isLaterThan(SimpleDate D)
   {
      < Write your implementation of this below. >
   }
   // Yields a random date
   public static SimpleDate random()
   {
      < Implementation assumed >
   }
}

public class Q4
{
    // Generates a random date and prints "yes" if it is later
    // in the year than June 15 and "no" otherwise.
    public static void main(String args[])
    {
      < Write your implementation of this below. >
    }
}
```

5. Here is a display of the new moon, first quarter moon, full moon, and third quarter moon:



Implement `drawQuarter` so that it performs as specified. Use `g.setColor(Color.black)` and `g.setColor(Color.yellow)` for setting the color of the "ink". Obtain the illuminated portion by drawing in yellow every possible horizontal line in the appropriate region.

```
public class ShowQuarters extends Frame
{
    // Graphics g draws a phased moon with radius r and center (hc,vc).
    // Depicts the first quarter moon if q = 1, the full moon if q = 2,
    // the third quarter moon if q=3, and the new moon otherwise.
    public static void drawQuarter(Graphics g,int hc, int vc, int r, int q)
    {
      < Write your implementation below. >
    }

    // Graphics g draws the four quarter phases of the Moon
    public void paint(Graphics g)
    {
       for(int k=0;k<4;k++)
          drawQuarter(g,200+150*k,400,50,k);
    }
}
```

For your information, the equation for a circle with center $(h_c\ , v_c)$ and radius r is given by

$$(h-h_c)^2 + (v-v_c)^2 \quad = \quad r^2$$

1.

```
20
10
XX
```

```
String c1 = s.substring(0,1);
String c2 = s.substring(1,2);
B = c1.equals(c2);
```

2.

Output:

```
 y = 11
 x = 11  y = 10  z = 30
 y = 2
 z = 4
 x = 1   y = 10  z = 4
```

3.

An implementation of nearlyPrime:

```
boolean twoOrThree = (n%2==0) || (n%3==0);
boolean fiveOrSeven = (n%5==0) || (n%7==0);
return twoOrThree && fiveOrSeven;
```

An implementation of main:

```
int sum = 0;
for (int k=1;k<=1000000;k++)
{
   if (A.nearlyPrime(k))
   {
      sum = sum + k;
   }
}
System.out.println(sum);
```

4.
An implementation of `isLaterThan`:

```
if (m>D.m)
   return true;
else if (D.m > m)
   return false;
else if (d > D.d)
   return true;
else
   return false;
```

Alternative:

```
   return ( (m>D.m) || ((m==D.m) && (d>D.d)) );
```

An implementation of `main`:

```
SimpleDate June15 = new SimpleDate(6,15);
SimpleDate D0 = SimpleDate.random();
if (D0.isLaterThan(June15))
   System.out.println("Yes");
else
   System.out.println("No");
```

5. An implementation of drawQuarter:

```
g.setColor(Color.black);
g.fillOval(hc-r,vc-r,2*r,2*r);
g.setColor(Color.yellow);
int d;
for(int v=vc-r;v<=vc+r;v++)
{
   d = (int) Math.sqrt(r*r - (v-vc)*(v-vc));
   if (q==1)
      g.drawLine(hc,v,hc+d,v);
   else if (q==2)
      g.drawLine(hc-d,v,hc+d,v);
   else if (q==3)
      g.drawLine(hc-d,v,hc,v);
}
```