CIS 180 – Object-Oriented Programming I
Midterm Examination #2

Name _____ ID _____

1. (25 pts) Give the output of the following program in the spaces provided:

```
public class A {
   public static void main(String[] args) {
      int x = 12;
      int y = 20;

      C c1 = new C();
      B b1 = new B();
      B b2 = new B();
      y = c1.m1(b1, y);
      System.out.println(x + ", " + y); // 12, 20
      c1.print();    // 20, 10, 1
      b1.print();    // 10, 2, 3
      b2.print();    // 1, 2, 3
   }
}


public class B {
   private int x = 1;
   private int y = 2;
   private int z = 3;

   public void setX(int i) {
       x = i;
   }

   public int getX() {
      return x;
   }

   public void print() {
      System.out.println(x + ", " + y + ", " + z);
   }
}
```

```
public class C {
   private int x = 5;
   private int y = 10;
   private int z = 15;

   public int m1(B b, int x) {
      int y = x;
      z = b.getX();
      this.x = y;
      b.setX(this.y);

      return y;
   }

   public void print() {
      System.out.println(x + ", " + y + ", " + z);
   }
}
```

2. A. (10 pts) Reformat the body of the method below to use proper indentation. **Do not change anything except the indentation.**

```
int f(boolean p, boolean q)
{
  if (p && !q)
  return 1;
  else if (p)
  return 2;
  else if (q)
  return 3;
  if (!p)
  if (q)
  return 4;
  else if (p && q)
  return 5;
  return 6;
}
```

```
int f(boolean p, boolean q)
{
   if (p && !q)
      return 1;
   else if (p)
      return 2;
   else if (q)
      return 3;
   if (!p)
      if (q)
         return 4;
      else if (p && q)
         return 5;
   return 6;
}
```

B. (10 pts) Complete the table below to show values returned from method f for various values of arguments p and q:

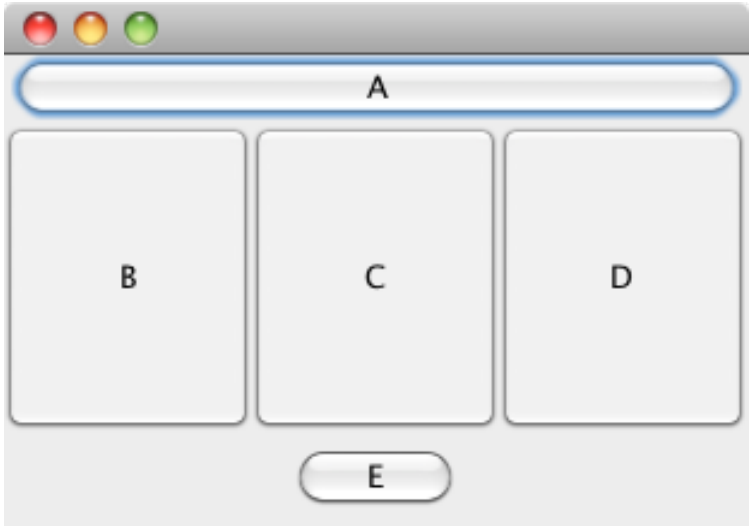
p	q	Return Value
true	true	**2**
true	false	**1**
false	true	**3**
false	false	**6**

C. (5 pts) Write a new version of method f to return values according to the following table.

p	q	Return Value
true	false	1
false	true	2
true	true	3
false	false	4

```
int f(boolean p, boolean q)
{
   if (p && !q)
      return 1;
   else if (!p && q)
      return 2;
   else if (p && q)
      return 3;
   else
      return 4;
}
```

3. (25pts) Complete the constructor method of the class given below to layout the buttons in the following arrangement:



Button A keeps its natural height and stretches horizontally to fill the width of the panel. Buttons B, C, and D each stretch horizontally to fill 1/3 of the width and vertically to fill the space between buttons A and E. Button E keeps its natural width and height and remains centered under button C as the panel is resized.

```
public class ButtonPanel extends JPanel
{
  private JButton a = new JButton("A");
  private JButton b = new JButton("B");
  private JButton c = new JButton("C");
  private JButton d = new JButton("D");
  private JButton e = new JButton("E");

  public ButtonPanel()
  {
     this.setLayout(new BorderLayout());

     JPanel bcdPanel = new JPanel();
     bcdPanel.setLayout(new GridLayout(1,3));
     bcdPanel.add(b);
     bcdPanel.add(c);
     bcdPanel.add(d);

     JPanel ePanel = new JPanel();
     ePanel.add(e);

     this.add(a, BorderLayout.NORTH);
     this.add(bcdPanel, BorderLayout.CENTER);
     this.add(ePanel, BorderLayout.SOUTH);

  }
}
```

4. (25 pts) Consider the class definitions below. Course objects maintain information about courses and have accessor methods for getting and setting the course title, number, and credit hours.

   A CourseInfoEditor should display information about a course, and allow the user to edit the information. The code to display the user interface has been completed for you. Fill in the blanks to complete the CourseInfoEditor class. Complete the constructor to display the course information in the text fields and add code to update the course object when the OK button is clicked.

```
public class Course {
   private String title;
   private String number; // e.g. "CIS-180"
   private int credits; // credit hours

   public int getCredits() {
      return credits;
   }

   public void setCredits(int credits) {
      this.credits = credits;
   }

   public String getNumber() {
      return number;
   }

   public void setNumber(String number) {
      this.number = number;
   }

   public String getTitle() {
      return title;
   }

   public void setTitle(String title) {
      this.title = title;
   }
}
```

```
public class CourseInfoEditor extends JFrame
   implements ActionListener {
   private Course course;
   private JTextField title = new JTextField(30);
   private JTextField number = new JTextField(8);
   private JTextField credits = new JTextField(2);
   private JButton ok = new JButton("OK");

   public CourseInfoEditor(Course c) {
      super("Course Info");
      course = c;

      this.setLayout(new FlowLayout());
      this.add(new JLabel("Title: "));
      this.add(title);
      this.add(new JLabel("Course Number: "));
      this.add(number);
      this.add(new JLabel("Credit Hours: "));
      this.add(credits);
      this.add(ok);

      title.setText(c.getTitle());
      number.setText(c.getNumber());
      int numCredits = c.getCredits();
      credits.setText(String.valueOf(numCredits));

      ok.addActionListener(this);

      this.setSize(450, 100);
      this.setVisible(true);
   }

   public void actionPerformed(ActionEvent e) {
      course.setNumber(number.getText());
      course.setTitle(title.getText());
      course.setCredits(
         Integer.parseInt(credits.getText()));
   }
}
```