Assignment 6

CSC 4330
Fall 2009

David Knowles

1a)
```
    System.out.print("Contents of v: ");
    Iterator i = v.iterator();
    while (i.hasNext()) {
        System.out.print(i.next() + ", ");
    }
    System.out.println();
```
1b)
```
    System.out.print("Contents of v: ");
    Enumeration e = v.elements();
    while (e.hasMoreElements()) {
        System.out.print(e.nextElement() + ", ");
    }
    System.out.println();
```

2a) 7
 b) 3
 c) 4
 d) 4
 e) 7
 f) 3

3)
```
    #include <iostream>
    template <class T>
    void myswap(T &a, T &b)
    {
      T temp = a;
      a = b;
      b = temp;
    }

    int main()
    {
      using namespace std;
      int a[2] = {1, 2};
      int i = 0;
      myswap(a[i], a[++i]);
    }
```

4a)

- **Generics** allow a method to operate on a variety of object types while still being type safe. i.e. allowing objects that implement the Serializable interface.
- **Autoboxing** allows Java to automatically convert primitive types into primitive wrapper classes. i.e. turning a literal "4" into an Integer object with value of 4.

4b)

Java checks the format strings against the actual arguments passed in to ensure they are compatible, and if they don't an **IllegalFormatException** is thrown. C doesn't do any checking and blindly attempts to do the conversions, which may or may not work as expected if an invalid argument type is passed in.

5)

```
function Sumer(Mat, Rows, Cols) result(Sum)
  implicit none
  integer, intent(in) :: Rows, Cols
  real :: Mat(Rows, Cols)
  real :: Sum

  integer Row, Col
  Sum = 0.0
  do Row = 1, Rows
     do Col = 1, Cols
        Sum = Sum + Mat(Row, Col)
     end do
  end do

end function Sumer

program Main
  real Mat_1(2, 3)
  Mat_1 = reshape( (/ 1.0, 1.0, 1.0, 1.0, 1.0, 2.0 /), (/ 2, 3 /) )

  print *, Sumer(Mat_1, 2, 3)
end program Main
```

6)
```
    #include <iostream>
    using namespace std;
    typedef void (*function)();
    int x;

    void foo()
    {
      cout << x << endl;
    }

    void bar(function sub)
    {
      int x = 3;
      sub();
    }

    void baz()
    {
      int x = 2;
      bar(foo);
    }

    int main()
    {
      x = 1;
      baz();
    }
```