1. (35 points)

   (a) Give the output for the following program that compiles and executes without error or warning.

   (b) On line #16 the programmer is pushing an integer onto the vector vec. However, vec contains instances of Android (see line #14). How does line #16 work?

```
1  #include <iostream>
2  #include <vector>
3  class Android {
4  public:
5    Android() { std::cout << "default" << std::endl; }
6    Android(int) { std::cout << "convert" << std::endl; }
7    Android(const Android&) { std::cout << "copy" << std::endl; }
8    Android& operator=(const Android&) {
9      std::cout << "assign" << std::endl;
10     return *this;
11   }
12 };
13 int main() {
14   std::vector<Android> vec;
15   for (unsigned int i = 0; i < 2; ++i) {
16     vec.push_back( i );
17   }
18 }
```

convert
copy
convert
copy
copy
****
Line #16 works because the conversion constructor is called, automatically by the compiler, to make an Android.

---

2. (5 points) Write a four letter synonym for the word value, as in " pass by value".
   copy

3. (20 points)

   (a) Give the output for the following program.

   (b) What's the biggest improvement that you could make to line #14.

```
1  #include <iostream>
2  class Android {
3  public:
4    Android() { std::cout << "default" << std::endl; }
5    Android(int) { std::cout << "convert" << std::endl; }
6    Android(const Android&) { std::cout << "copy" << std::endl; }
7  };
8  class CyberLife {
9  public:
10   CyberLife(const Android& p) { mon = p; }
11 private:
12   Android mon;
13 };
14 int main() {
15   CyberLife connor(800);
16 }
```

```
convert
default
assign
**********************
```

Change line #14 so that it uses an initialization list:

```
CyberLife(const Android& p) : mon(p) { }
```

---

4. (5 points) Give the output for the following program.

```
1  #include <iostream>
2  #include <cstring>
3
4  class string {
5  public:
6    string(const char * s) : buf(new char[strlen(s)+1]) {
7      strcpy(buf, s);
8    }
9    char* getBuf() const { return buf; }
10 private:
11   char *buf;
12 };
13
14 int main() {
15   string x("cat");
16   char* buf = x.getBuf();
17   buf[0] = 'r';
18   std::cout << x.getBuf() << std::endl;
19 }
```

```
rat
```

5. (25 points) Give output for the following program that compiles and executes without error or warning.

```
#include <iostream>
#include <cstring>
class string {
public:
  string() { std::cout << "default" << std::endl; }
  string(const char*) { std::cout << "convert" << std::endl; }
  string(const string&) { std::cout << "copy" << std::endl; }
  ~string() { std::cout << "destructor" << std::endl; }

  string& operator=(const string&) {
    std::cout << "copy assign" << std::endl;
    return *this;
  }
};

void fun(string) { }

int main() {
  string cat("cat"), dog = cat;
  fun(cat);
  int x = 17;
  int& r = x;
  r = 99;
  std::cout << x << std::endl;
}
```

convert
copy
copy
destructor
99
destructor
destructor

---

6. (10 points) Write a copy assignment operator for class Pokemon.

```
#include <iostream>
#include <cstring>

class Pokemon {
public:
  Pokemon& operator=(const Pokemon& rhs) {
    if ( this == &rhs ) return *this;
    delete [] name;
    name = new char[strlen(rhs.name)+1];
    strcpy(name, rhs.name);
    return *this;
  }
private:
  char* name;
};
```