1. (10 points) Give the output for the following program.

```
#include <iostream>
2 int cout = 45\%7?5:0;
3 int end1 = 7/2;
   namespace Mine {
5
     namespace std {
        float cout = 2/4;
6
7
8
  }
9
   int main() {
     std::cout << cout+endl << std::endl;</pre>
      std::cout << Mine::std::cout << std::endl;
12
   8
   0
```

2. (15 points) Give the output for the following program.

```
#include <iostream>
   class Pokemon {
   public:
      Pokemon() { std::cout << "default" << std::endl; }
      Pokemon(int) { std::cout << "convert" << std::endl; }</pre>
      Pokemon(const Pokemon&) { std::cout << "copy" << std::endl; }
6
      Pokemon& operator = (const Pokemon&) {
        std::cout << "assign" << std::endl;</pre>
8
9
        return *this;
10
11
   private:
12
    int cp;
13
   class Pokedex {
15
   public:
     Pokedex (const Pokemon& p) { mon = p; }
16
17
   private:
18
     Pokemon mon;
19
   };
20
   int main() {
21
    Pokedex dex(2750);
22 }
   convert
   default
   assign
```

3. (5 points) Write a C++ program that generates and prints a random number between 5 and 10, including the end points.

```
1 #include <iostream>
2 int main() {
3    std::cout << rand()%6+5<< std::endl;
4 }</pre>
```

4. (10 points) Give the output for the following program.

```
#include <iostream>
   #include <vector>
   class Pokemon {
   public:
     Pokemon() { std::cout << "default" << std::endl; }</pre>
     Pokemon(int) { std::cout << "convert" << std::endl; }
     Pokemon(const Pokemon&) { std::cout << "copy" << std::endl; }
7
     Pokemon& operator = (const Pokemon&) {
        std::cout << "assign" << std::endl;</pre>
10
        return *this;
11
12
   private:
13
    int cp;
14
15
   int main() {
   std::vector < Pokemon > vec;
     for (unsigned int i = 0; i < 2; ++i) {
17
18
        vec.push_back( i );
19
20
   }
   convert
   copy
   convert
   сору
   сору
```

5. (10 points) Give the output for the following program. Note the use of reserve.

```
#include <iostream>
2 #include <vector>
   class Pokemon {
   public:
      Pokemon() { std::cout << "default" << std::endl; }
      Pokemon(int) { std::cout << "convert" << std::endl; }
6
      Pokemon(const Pokemon&) { std::cout << "copy" << std::endl; }
7
     Pokemon& operator = (const Pokemon&) {
8
9
        std::cout << "assign" << std::endl;</pre>
10
        return *this;
11
     }
12
   private:
    int cp;
13
14
   };
   int main() {
15
    std::vector < Pokemon > vec;
16
17
     vec.reserve(2);
     for (unsigned int i = 0; i < 2; ++i) {
19
        vec.push_back( i );
20
      }
21 }
   convert
   сору
   convert
   сору
```

6. (15 points) Give the output for the following program. After printing the output the program crashes with a double free error. To fix the error, what function should the programmer write?

```
#include <iostream>
2 #include <cstring>
3
4
   class string {
5
    public:
      string (const char * s) : buf(new char[strlen(s)+1]) {
7
        strcpy(buf, s);
        std::cout << "convert" << std::endl;</pre>
8
9
10
      ~ string() {
        delete [] buf;
11
        std::cout << "destructor" << std::endl;</pre>
12
13
14
      char* getBuf() const { return buf; }
15
      string& operator = (const string&) {
16
        std::cout << "assign" << std::endl;</pre>
17
        return *this;
18
19
    private:
20
     char *buf;
21
22
23
   int main() {
24
      string x("cat"), y = x;
25
      char* buf = x.getBuf();
26
      buf[0] = 'r';
      std::cout << x.getBuf() << std::endl;</pre>
27
28 }
    convert
   rat
   destructor
   *** Error in './run': double free or corruption
   Need a copy constructor
```

7. (5 points) Give the output for the following program.

```
#include <iostream>
2 class Pokemon {
   public:
     Pokemon(int p) : cp(p) { std::cout << "convert" << std::endl; }
5
     Pokemon& operator = (const Pokemon& p) {
6
       cp = p.cp;
        std::cout << "assign" << std::endl;</pre>
7
8
       return *this;
9
10
   private:
11
   int cp;
12 };
13
   int main() {
     Pokemon p(2500), q = p;
15 }
```

convert

8. (30 points) Write a copy constructor, assignment operator, and an output operator for the following program.

```
#include <iostream>
   #include < cstring >
3
4
   class Pokemon {
   public:
     Pokemon(const char* n) : name(new char[strlen(n)+1]) {
6
7
        strcpy (name, n);
8
     Pokemon(const Pokemon& p) : name(new char[strlen(p.name)+1]) {
9
10
        strcpy(name, p.name);
11
12
     Pokemon& operator = (const Pokemon& rhs) {
13
        if ( this == &rhs ) return *this;
14
        delete [] name;
15
        name = new char[strlen(rhs.name)+1];
16
        strcpy(name, rhs.name);
17
        return *this;
18
19
      const char* getName() const { return name; }
20
   private:
     char* name;
21
23
   std::ostream& operator <<(std::ostream& out, const Pokemon& p) {
24
      return out << p.getName();</pre>
25
   }
26
27
   int main() {
     Pokemon p("snorlax");
      std::cout << p << std::endl;
30 }
```