CSE130 - Dr. Roman V. Yampolskiy
Introduction to Programming Languages
Examination 2

Please Print

Name: Richard Douglas Section: cse130 User ID: 5540894

1. Books, Notes, Calculators, AI, etc. are not allowed.

2. You may not interact with anyone except the instructor during the examination.

3. You may not share any materials, offices supplies, etc. with anyone.

<u>Problem 1</u> (6 points)**:**

For each of the following, write T if the statement is true and F if it is false. You must use T or F. Do not use x's or check marks they will be counted as wrong answers.

1	OOP stands for Object Only Property.	F
2	Polymorphism means that each object can be used in more than one program.	F
3	Encapsulation means that memory is dynamically allocated.	**F**
4	The purpose of a constructor is to allocate memory to class instances.	**T**
5	To make a function an inline function place the reserved word inline before the function name, and define the function before any calls are made to it.	**T**
6	Inheritance makes it possible to use code written in a different language.	**F**

<u>Problem 2:</u> (12 points)

1. Suppose we've defined a book class to include a setTitle method with a prototype:

void setTitle(char *);

Suppose further we have an instance of the book class named bookOne. Determine which of the following statements correctly invokes the setTitle method: **<u>C</u>**

a. title = BookOne.setTitle("Book One");
b. title = book.setTitle("Book One");
<u>c. bookOne.setTitle("Book One");</u>
d. book.setTitle("Book One");

2. Let v be a static variable defined in a function called f

a. v gets initialized every time f is called
b. v can only be accessed once in the program
<u>c. v retains its value between calls of f</u>
d. v stays constants for all calls of f

3. Given the C++ declaration: class A {public: int x; protected: int y;};
Which of the following would be rejected by the compiler?

a. class B: public A { void f() { x = y; }};
<u>b. class B { void f() { A a; a.x = a.y; }};</u>
c. class B: public A{}; class C: public B { void f() { x = y; }};
d. None of the above

4. The output of the C++ code,

```
#include <iostream>
using namespace std;
class A { public: int f(int x) { cout << x << " "; }};
class B: public A { public: int f(int y) { A::f(y+1); }};
void g(A a, B b) { a.f(3); b.f(3); }
int main() { B p; B q; g(p,q); }
```

would be

a. 3 3
b. 3 4
c. 4 4
d. **None of the above** – won’t compile. No return on non-void functions.

5. Given the C++ declaration: template<class T> class set { ...}
which of the following declarations (outside of the template) could not be correct?

a. set s;
b. set<int> s;
c. set<float> s;
d. set< set<int> > s;

6. Which one of the following statements is NOT true about multiple inheritance in C++?

a. It allows a class to be considered a subclass of two other classes.
b. It allows objects of the child class to have all the members of two parent classes.
c. If a class is derived from two parent classes and both have a method with the same signature, the compiler will generate an error message.
d. A class can be derived from more than two classes.

Problem 3 (3 points)
Turn the C++ definition: int sum(int a, int b, int c) { return a + b + c; }
into a function template that can be used to work on any type that supports +, instead of just int.

template <class T>

T sum(T a, T b, T c) { return a + b + c }

Problem 4 (10 points)
Correct the syntax errors in each code segment below. If the code segment contains no error in the code segment, clearly write ***no error*** next to the code segment. ***Correct syntax errors only (that means only the errors that will prevent the code from compiling)***. Assume that all variables used have been declared.

```
cout << “Hello Worlds!”;          no error
```

```
cin >> x;
```

```
cout << “The sum of x and y is” << x + y << “integer units”;
```

```
int x = 10;
int y = 20;
x = y/10;
```

```
int x = 0;
int y = 10;
int z = x/y; //It would probably compile but zero division won’t work. ‘y’
as denominator would allow the program to actually run.
```

```
int x = 0;
cin >> y;
if (x == y) cout << “They are equal!!” << endl;
else{
   cout << “Doesn’t matter which comes first!” << endl;
   cout << “They are not equal!!!”;
}
```

```
if (x == y)
   cout << “X and Y are equal!” << endl;
else cout << “They are not equal.” << endl;
```

```
x = 0;
while (x < 10){
  x++;
  cout << “The value of x is: ” << x;
}
```

```
x = (y == 2);                  no error
```

```
(x + y) = z; ?? what do you even want here? Z = (x+y) will compile
```

<u>Problem 5</u> (3 points)
Give an example of an abstract class declaration:

```
class Shape {
public:
        Shape( double x, y);
        virtual getArea();
        virtual printArea() = 0;
private:
        double _x, _y;
};
```

<u>Problem 6</u> (6 points)
Please provide definitions for the overloaded swap() function, which swaps 2 values:

```
//can also just be overloaded with a template
template <class T>
void swap(T *a, T *b){
        T temp = *a;
        *a = *b;
        *b = temp;
}

void swap (int *a, int *b) {
        int temp = *a;
        *a = *b;
        *b = temp;
};
void swap (float *c, float *d) {
        float temp = *c;
        *c = *d;
        *d = temp;
};
void swap (char *p, char *q) {
        char temp = *p;
        *p = *q;
        *q = temp;
};
```

Problem 7: (10 points) Create a class named Point that represents points in a three dimensional space that has the following properties:

1. It contains three double variables x, y, z representing the components as instance variables.
2. It includes a default constructor that initialize x, y, and z to 0.
3. It includes a constructor that requires three arguments one for each component.
4. It includes a method to translate the point by dx, dy, and dz in the x, y, and z direction. That is new x component should become x + dx, etc.

```
class Point {
public:
        Point( double x = 0, double y = 0; double z = 0 );
        Point( double _x,_y,_z ){
                x=_x;
                y=_y;
                z=_z;
        };
        translate(double dx,dy,dz){
                x += dx;
                y += dy;
                z += dz;
        };

private:
        double x,y,z;
```