EDITH COWAN UNIVERSITY PERTH WESTERN AUSTRALIA		**INTERNAL/EXTERNAL** Semester One, 2016
Unit Code and Title	**CSP2104 Object-Oriented Programming in C++**	**STANDARD PAPER**

Duration	Reading time 5 minutes Working time 3 hours Total time 3 hours 5 minutes
Attempt	All questions
Marks	As shown on question, 24 marks in total
Type of Exam	**Closed Book** exam – unit guide, text or reader are not permitted

Special Instructions

- This examination paper consists of 1 part.
- There are a total of 11 pages.
- *Some exam papers are double sided – please check both sides of the page*

Students are not permitted to write on the examination or any other paper during reading time.

Do not commence the examination until you are told to do so.

Question 1 (3 Marks)

For the following program, modify only the line indicated by the arrow so that for loop prints characters from the *secretMessage* array in the following pattern: **jgda**.

```
#include "stdafx.h"
#include<iostream>
using namespace std;
char secretMessage[12] = {'a','b','c','d','e','f','g','h','i','j','k','l' };

int main()
{
	for (__________;__________;__________)   ⬅
	{
		cout << secretMessage[i];
	}
	system("pause");
	return 0;
}
```

Question 2 (2 Marks)

The output (to screen) of the following code snippet is: ______________________.

```
	struct Stuff
	{
		int x;
	};

	Stuff myStuff;
	vector<Stuff> someStuff;
	myStuff.x = 1;
	someStuff.push_back(myStuff);
	myStuff.x = 2;
	someStuff.push_back(myStuff);

	Stuff otherStuff;
	vector<Stuff*> collection;
	otherStuff.x = 3;
	collection.push_back(&otherStuff);
	otherStuff.x = 4;
	collection.push_back(&otherStuff);

	cout << someStuff[0].x;
	cout << someStuff[1].x;
	cout << ", ";
	cout << collection[0]->x;
	cout << collection[1]->x;
	system("pause");
```

Question 3 (4 marks)

The output (to screen) of the following code snippet is: ______________________.

```
float compute(float a, float *b, float &c)
{
	a += 2;
	(*b)--;
	c++;
	return(a + *b + c);
}

int main()
{
	float x, y, z;
	x = 8; y = 4; z = 0;
	float w = compute(x, &y, z);
	cout << x << "," << y << "," << z << "," << w << "\n";
	return 0
}
```

Question 4 (2 Marks)

Consider the Vector3 struct which is defined in the code below:

```
struct Vector3
{
	double x, y, z;
};
```

In the space provided below, write a function setVector which accepts four inputs: a reference to a Vector3 type variable, *beingSet*, and three individual double type numbers, *a*, *b* and *c*. The function should set the x, y and z components *beingSet* to *a*, *b* and *c* respectively.

When the setVector function is used as follows:

```
Vector3 oneVector;
setVector(oneVector, 1, 4, 5);
cout << oneVector.x << ", " << oneVector.y << ", " << oneVector.z << "\n";
```

The output should be: 1,4,5

Answer Space:

Question 5 (1 Mark)

Continuing the Vector3 struct example from Question 4, write a function which overloads the addition operator (+) for summing two Vector3 type variables, producing another Vector3 whose x, y and z components are sums of the individual x, y and z components of the vectors being summed.

i.e. Once correctly implemented, the following code:

```
Vector3 oneVector, otherVector;
setVector(oneVector, 1, 4, 5);
setVector(otherVector, 10, 11, 12);
Vector3 thirdVector = oneVector + otherVector;
cout << thirdVector.x << ", " << thirdVector.y << ", " << thirdVector.z << "\n";
```

Should produce the output: 11, 15, 17

Answer Space:

Question 6 (3 Marks)

In the space below, write the code to define and implement the following class:
Class name: DictionaryElement
Private member variables: string *word*, string *definition*
Public methods:

- Constructor that accepts two strings and sets *word* and *definition*.
- A getter method for *word*
- A getter method for *definition*

Answer Space:

Question 7 (3 Marks)

Assuming you have the DictionaryElement class from Question 6, use inheritance to define a *ThesaurusElement* class (inheriting from DictionaryElement), containing:
Private variable: vector of DictionaryElements called *similes*.
Public getter method for vector of similes.
Public addSimile method that adds an input DictionaryElement to the similes vector.

Answer Space:

Question 8 (3 Marks)

Again, assuming the DictionaryElement class from Question 6 exists – re-implement the *ThesaurusElement* class that you wrote in Question 7. Instead of using inheritance use composition. Your *ThesaurusElement* class should include a DictionaryElement to store a word and definition and a vector of DictionaryElements to store the similes.
Implement the constructor, getter method for similes and addSimile method so that this class can be used in the same way as the class you wrote for Question 7.

Answer Space:

Question 9 (2 Marks)

Consider the class defined in the code below:

```
class ConDestructFun
{
public:
	ConDestructFun()
	{
		objectID = objectCount;
		objectCount++;
		cout << "Object " << objectID << " created.\n";
	}
	~ConDestructFun()
	{
		cout << "Object " << objectID << " destroyed.\n";
	}
private:
	static int objectCount;
	int objectID;
};
int ConDestructFun::objectCount = 0;
```

What is printed to the console when the code below is run:

```
int main()
{
	{
		ConDestructFun a;
		{
			ConDestructFun b;
			{
				ConDestructFun c;
			}
		}
	}
	return 0
}
```

Answer Space (Console Output):

Question 10 (1 Mark)

Again, considering the ConDestructFun from Question 9, what is printed to the console when the code below is run:

```
int main()
{

	{
		ConDestructFun *a = new ConDestructFun();
		{
			ConDestructFun *b = new ConDestructFun();
			{
				ConDestructFun *c = new ConDestructFun();
			}
		}
	}
	return 0;
}
```

Answer Space (Console Output):

END OF EXAMINATION PAPER