Open Universities AUSTRALIA

Study Period 1

University	***RMIT University***
Exam Code(s):	***CPT323***
Title of Paper:	***Object Oriented Programming Using C++***
Exam Duration:	*2 hours*
Reading Time:	*10 minutes*

During an exam, you must not have in your possession, a book, notes, paper, electronic device(s), calculator, pencil case, mobile phone, smart watch/device or other material/item which has not been authorised for the exam or specifically permitted as noted below. Any material or item on your desk, chair or person will be deemed to be in your possession. You are reminded that possession of unauthorised materials in an exam is a discipline offence.

No examination papers are to be removed from the room.

Authorised Materials

Calculators	□ Yes	☑ No
Open Book	□ Yes	☑ No
Specifically Permitted Items	□ Yes	☑ No

If yes, specifically permitted items are:

Students must complete this section if required to write answers within this paper

OUA ID:
Provider University ID:
Family name:
Other names (in full):

Part A – Give a short answer to the following questions. (60 marks)

1. What is the type of the `std::cin` object? (1 mark)
2. Write a #include statement to include the `std::shared_ptr` class in your program. (1 mark)
3. Given that the following is valid c++ code:

   ```
   std::string s = “fred”;
   ```

   Does that mean that “`fred`” is a std::string? If it is not a std::string, what is it? Be as precise as you can (1 mark)
4. What is the default visibility of c++ class members? (1 marks)
5. If we have a common parent class A, and two child classes, B and C, how can I safely downcast from A to B or C in order to access methods only defined in one of the child classes? (1 mark)
6. Given that the STL `map` supports the [] operator for accessing elements, does this mean that it has vector / array like performance for accessing its elements? Why is/is not that the case? (3 marks)
7. Show an example of using the `std::stoi` operator to extract an integer from a string. Include any exception handling (2 marks)
8. Name the three classes introduced in c++11 that implement the safe pointer idea. (3 marks)
9. The following code generates a compiler error. What is the reason that this is a compiler error (not just the location but the actual reason that it is a compiler error)? You may assume that there is a validly defined widget class in scope.(1 mark)

   std::vector<std::vector<widget&> > widgets;
10. What problem exists in the following code? How could we avoid it? (3 marks)

    ```
    class foo
    {
            int i;
            public:
                    foo() : i(0) {}
    };
    class bar : public foo
    {
            friend ostream& operator<<(ostream&, const bar&);
    };
    ostream& operator<<(ostream& out, const bar& b)
    {
            out << b.i;
            return out;
    }
    ```
11. What does it mean to “inline” a function? How is this achieved and what benefit do we get from doing so. Is there a cost or down-side to doing this? If so, what is it? (5 marks)
12. What object-oriented concept is enabled through the use of the “virtual” keyword? Explain What benefit(s) does this give us? Are there any costs? If so, outline one of these. (4 marks)

13. In the C programming language, generic programming would be achieved using void pointers. C++ has introduced a better way to achieve this. What is that better way? Why is it better? Is there a runtime cost to this better way? Why/ why not? (6 marks)

14. What is the output of the following program? (2 marks)

```
#include <iostream>
#include <cstdlib>
#include <memory>
class base
{
	public:
		base()
		{
			std::cerr << "base created" << std::endl;
		}

		virtual ~base()
		{
			std::cerr << "base destroyed" << std::endl;
		}
};

class derived : public base
{
	public:
		derived()
		{
			std::cout << "derived created" << std::endl;
		}

		virtual ~derived()
		{
			std::cout << "derived destroyed" << std::endl;
		}
	};

int main(void)
{
	std::unique_ptr<derived>d = std::make_unique<derived>();
	return EXIT_SUCCESS;
}
```

15. There is a memory access problem in this code segment below. Identify the problem and a solution to the problem. Only rewrite those lines of code you need to to fix the problem. (6 marks)

```
#include <cstdlib>
#include <exception>
#include <iostream>
#include <sstream>
#include <vector>
class outofrange : public std::exception
{
        int offender_;
        public:
        outofrange(int offender) : offender_(offender) {}
        const char * what() const throw()
        {
                std::ostringstream oss;
                oss << offender_ << " is out of range";
                return oss.str().c_str();
        }
};
int main(void)
{
        std::vector<int>ints(40);
        /* some code to initialise the elements of the vector here */
        int input;
        std::cout << "Please enter an index to get the value of: " << std::endl;
        std::cin >> input;
        try
        {
                if(input > int(ints.size()))
                {
                        throw outofrange(input);
                }
        }
        catch(outofrange& oor)
        {
                std::cerr << oor.what() << std::endl;
        }
        return EXIT_SUCCESS;
}
```

16. Name three types of iterators are supported by the Standard Template Library. What types of iteration do each type of iterator allow? (6 marks)

17. Unlike more modern object-oriented languages such as java, c++ arrays of objects are stored contiguously in memory. What advantages does this provide in terms of performance. Name a modern hardware component that assists in this process and explain in a sentence or two how it helps. (8 marks)

18. There is a linker error and a compiler error in the following program. Identify the errors and how we might fix them. (6 marks)

```
in foo.h:

#include <iostream>
#include <cstdlib>
int i;
class foo
{
   int t;
   public:
          foo(int & _t) : t(_t) {}
          const int & get_t() const {
          return t; }
};
```

```
in foo.cpp:

#include “foo.h”
int i;
int main(void)
{
       foo f(3);
       std::cout << f.get_t() <<
       std::endl;
       std::cout << i << std::endl;
       return EXIT_SUCCESS;
}
```

Part B: Anatomy of a Class (60 Marks)

We wish to model bicycle parts for a company selling these. As part of this process we will use various advance features of c++ classes to keep track of information on each part. A bicycle part has a part number, a weight, a price, a brand name and a country of origin.

1. Write the declaration (not the implementation) of the class bicycle. Assume that this class is defined in a file called bicycle_part.h. If you feel you need to create other classes or structs to complete the functionality here, you should feel free to go ahead and do so. Include the prototypes for the following methods. Note that not all are required for this class – you are required to make these to show the concepts you have learnt in this course (8 marks):
    a. A default constructor.
    b. A constructor that initialises all elements specified in the above description.
    c. A copy constructor.
    d. The assignment operator for this class.
    e. A conversion operator to the 'double' type
    f. The prototype for a static member function called total_cost which takes in a vector of bicycle parts and returns the total cost for these parts.
    g. Make the operator<<() for printing out a bicycle_part a **friend** of this class with an appropriate declaration.
    h. The prototype for a destructor.

    In the following questions, you will be implementing the components of the class that were specified above. Each of these members should be assumed to be implemented in the .cpp file for this class – called bicycle.cpp. Note that if you need to create additional functions to make your code work, go ahead and implement those but they should be short helper functions.

2. Implement the default constructor specified in 1 a) to initialize all values to sensible defaults. (6 marks)
3. Implement the parameterized constructor specified in 1 b). (6 marks)
4. Implement the copy constructor specified in 1 c).(6 marks)
5. Implement the assignment operator specified in 1 d) (6 marks)
6. Implement the conversion operator specified in 1 e). This conversion operator should simply return the weight of the bike part. (8 marks)
7. Implement the static member function specified in 1 f).(10 marks)
8. Implement the operator<<() function specified in 1 g). It should output the values for this bicycle parts separated by commas. (8 marks)
9. Implement the destructor specified in 1 h). Note: only do what is absolutely necessary here. (2 marks)