UNIVERSITY OF SWAZILAND

FACULTY OF SCIENCE & ENGINEERING

DEPARTMENT OF ELECTRICAL & ELECTRONIC ENGINEERING

RESIT EXAMINATION

July 2018

TITLE OF PAPER: PROGRAMMING TECHNIQUES II

COURSE CODE: EEE272/EE272

DURATION: 3 HOURS

INSTRUCTIONS:

- 1. There are five (5) questions in this paper. Answer question 1 and any other three (3) questions.
- 2. Each question carries equal marks.
- 3. Use correct notation and show all your steps clearly in any program analysis.
- 4. All programs should be sufficiently commented and indented for clarity.
- 5. Start each question in a new page.

This paper should not be opened until permission has been given by the invigilator.

This paper contains four (4) pages including this page.

٠

Question 1

	a.	How are overloaded functions differentiated from each other when a program is		
		executed?	[2]	
	b.	Define the principles of encapsulation and information hiding in object oriented		
		programming. How does the definition of classes promote these principles?	[4]	
	c.	Explain why a class might provide set and get functions to update and access data		
		members.	[2]	
	d.	What is the purpose of access specifiers? Explain the access control level provided		
		each access specifier.	[6]	
e. Explain why data members, in particular should be placed in the private			a	
		class.	[2]	
	f. By using an example explain the difference between a base class and a derived class			
			[2]	
	g.	What is a default class constructor? Describe two ways by which it can be created.[4]		
	h.	What is the role of a destructor in a class?	[3]	
Qu	est	tion 2		
	a.	Explain the difference between function overloading and overriding.	[2]	

 b. Provide the syntax to declare a C++ class B to be a public base class for derived class D. [2] c. What is this pointer? By way of illustration give two instances where you may be required to use this pointer. [5] d. Explain the following C++ concepts; i. friendship ii. Operator overloading iii. Public inheritance e. What is the rationale behind separating class interface from implementation? [2] f. Why is it that operator<< cannot be overloaded as a member function? [2] g. How is a postfix unary operator++ differentiated from the prefix version? [2] h. What is the difference between class composition and inheritance? [2] i. What are some reasons for using friend operator overloading? [2] 			[-]
 c. What is this pointer? By way of illustration give two instances where you may be required to use this pointer. d. Explain the following C++ concepts; i. friendship ii. Operator overloading iii. Public inheritance e. What is the rationale behind separating class interface from implementation? [2] f. Why is it that operator<< cannot be overloaded as a member function? [2] g. How is a postfix unary operator++ differentiated from the prefix version? [2] h. What is the difference between class composition and inheritance? [2] i. What are some reasons for using friend operator overloading? 	b.	Provide the syntax to declare a C^{++} class B to be a public base class for derive class D .	d [2]
 d. Explain the following C++ concepts; friendship Operator overloading Public inheritance e. What is the rationale behind separating class interface from implementation? [2] f. Why is it that operator<< cannot be overloaded as a member function? [2] g. How is a postfix unary operator++ differentiated from the prefix version? [2] h. What is the difference between class composition and inheritance? [2] i. What are some reasons for using friend operator overloading? [2] 	c.	What is this pointer? By way of illustration give two instances where you may be required to use this pointer.	[5]
 e. What is the rationale behind separating class interface from implementation? [2] f. Why is it that operator<< cannot be overloaded as a member function? [2] g. How is a postfix unary operator++ differentiated from the prefix version? [2] h. What is the difference between class composition and inheritance? [2] i. What are some reasons for using friend operator overloading? [2] 	d.	Explain the following C++ concepts; i. friendship ii. Operator overloading iii. Public inheritance	[6]
 f. Why is it that operator<< cannot be overloaded as a member function? [2] g. How is a postfix unary operator++ differentiated from the prefix version? [2] h. What is the difference between class composition and inheritance? [2] i. What are some reasons for using friend operator overloading? [2] 	e.	What is the rationale behind separating class interface from implementation?	[2]
 g. How is a postfix unary operator++ differentiated from the prefix version? [2] h. What is the difference between class composition and inheritance? [2] i. What are some reasons for using friend operator overloading? [2] 	f.	Why is it that operator << cannot be overloaded as a member function?	[2]
 h. What is the difference between class composition and inheritance? [2] i. What are some reasons for using friend operator overloading? [2] 	g.	How is a postfix unary operator++ differentiated from the prefix version?	[2]
i. What are some reasons for using friend operator overloading? [2]	h.	What is the difference between class composition and inheritance?	[2]
	i.	What are some reasons for using friend operator overloading?	[2]

j. What is class composition? Give an example of class composition. [3]

Question 3

a. Based on the shown member function prototype, describe how the member function operates. [5]

int get(const char&, const int)_const;

- b. In the class it was mentioned that the operator+ for an expression such as 10 + Simple(3) cannot be overloaded as a member function. Why it so? Write a simple code that includes a class with data members and member functions to illustrate how the operator could be appropriately overloaded for the given expression. [10]
- c. When a derived class inherits from a base class, how is the base class constructor called? [3]
- d. Write statements for creating a dynamic array that stores 10 integer elements. [3]
- e. Suppose we have a class D derived from base class B,

```
class B
ł
public:
  // other members
  void func();
// other members
};
void B::func() { /* body */}
class D: public B
public:
   // other members
  void func();
  // other members
};
void D::func() { /* body */}
int main() {
D dObj;
//other statements
}
```

Make the following call using dObj as calling object:

dObj.func();

Which version of func is called? Explain your answer. [4]

Question 4

Write a complete C++ program to do the following:

- Student is a base class, having two data members: entryno and name; entryno is an integer and name is of type characters, 20 characters long. The value of entryno is 1 for Science student and 2 for Arts student, otherwise it is an error.
- Science and Arts are two derived classes, having respectively data items marks for science and marks for arts.
- Read appropriate data from the keyboard for 3 science and 2 arts students.
- The two derived classes have members function display which is used to display entryno, name, marks for science students first and then for arts students.

i.	Write the C++ interface.	[5]
ii.	Write the C++ implementation.	[15]
iii.	Write a C++ driver program for the class	[5]

Question 5.

Write a complete C^{++} program that uses class rectangle and point. Class rectangle only stores Cartesian coordinates of type point for the four corners of the rectangle. The class must include a set function that does the following:

- Before assigning the sets of coordinates to data members, it must verify that they are in the first quadrant with no single x or y coordinate larger than 20.0.
- It must also verify that the supplied coordinates specify a rectangle.

Other member functions include:

- A constructor that uses the set function to initialise the coordinates.
- Member functions to calculate length, width, perimeter and area.
- A member function which determines whether the rectangle is a square.

i.	Write the C++ interface.	[5]
ii.	Write the C++ implementation.	[15]
iii.	Write a C++ driver program for the class	[5]

End of paper