

Name:

Enrolment No:



UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

End Semester Examination, May 2019

Programme Name: B.Tech. ECE
Course Name : OOPs with C++
Course Code : CSEG 2001
Nos. of page(s) : 9
Instructions:

Semester : IV
Time : 03 hrs
Max. Marks: 100

SECTION A [20 Marks]

S. No.		Marks	CO
Q 1.	<p>I. Which of the following operators can use friend functions for overloading?</p> <p>a. == b. [] c. -> d. ()</p> <p>II. Fill in the blank:</p> <pre>#include<iostream> using namespace std; class Test { static int x; public: void get() { x = 15; } void print() { x = x + 20; cout << "x =" << x << endl; } }; _____ ; // Define static variable 'x'</pre> <pre>int main() { Test o1, o2; o1.get(); o2.get(); o1.print(); o2.print(); return 0; }</pre>	2+2	CO1, CO2

	}		
Q 2.	<p>Try to fetch the expected output for the following code:</p> <pre> #include <iostream> using namespace std; class Test { int i; public: Test(int ii) : i(ii) {} const Test operator*(const Test& rv) { cout << "Executes *" << endl; return Test(i * rv.i); } Test& operator+=(const Test& rv) { cout << "Executes +=" << endl; i += rv.i; return *this; } }; int main() { int i = 1, j = 2, k = 3; k += i * j; Test ii(1), jj(2), kk(3); kk += ii * jj; return 0; } </pre>	4	CO2
Q 3.	<p>Consider the following code segment. Assume that the sizeof(int) = 4 and sizeof(double) = 8.</p> <pre> class base { static int statInt; double arr[5]; void display() {} }; class base1: public base { }; class base2: public base { }; class derived: protected base1, protected base2 { }; </pre> <p>What will be the size of the object derived?</p>	4	CO3

	<p>a) 84 b) 120 c) 80 d) 44</p>		
Q 4.	<p>What will be the output of the following program?</p> <pre> #include <iostream> using namespace std; class Test { public: Test() { cout << "Created" << endl; } ~Test() { cout << "Destroyed " << endl; } }; int main() { try { Test t1; throw 98; } catch(char i) { cout << "Caught Char " << i << endl; } catch(double i) { cout << "Caught Double " << i << endl; } catch(...) { cout << "Default" << endl; } return 0; } </pre>	4	CO1, CO2
Q5.	<p>What is the output of the following code?</p> <pre> #include <iostream> using namespace std ; namespace Ex { int x = 10; } namespace Ex { int y = 10; } int x = 5; </pre>	4	CO1

```
int main()
{ using namespace Ex ;
x = y = 50;
cout << x << " " << y;
return 0;
}
```

- a. 10 10
- b. 50 50
- c. 5 50
- d. Compilation error: ambiguous reference to variable 'x'

SECTION B [40 Marks]

Q 6.

- (a) Which member functions are created automatically by the compiler if they are not included (by Programmer) in the class definition?
- (b) This program is all about the implementation of Pre/Post Incrementer. Fill the blank by keeping this in mind so that the given test cases will satisfy:

```
#include <iostream>
using namespace std;
```

class MyClass

```
{ int data;
public:
_____ {} // Define Constructor
```

```
MyClass& operator++()
{ ++data;
return _____;
}
```

```
_____ {
MyClass t(data);
++data;
return _____; }

```

```
void disp()
{ cout << " " << data ; }
};
```

```
int main()
{
int x;
```

5+5

**CO2,
CO1**

	<pre> cin >> x; MyClass obj1(x); obj1.disp(); MyClass obj2 = obj1++; obj2.disp(); obj2 = ++obj1; obj2.disp(); return 0; } </pre>		
Q 7.	<p>Write a program in C++ to overload the operator “new” and “delete”. Clearly define their definition utilizing malloc() and free() function.</p>	10	CO3
Q 8. (a)	<p>What will be the output of the following program?</p> <pre> #include <iostream> #include <string> #include <iostream> #include<string> using namespace std; class Department { public: string dept; Department(string d):dept(d) {} void getDeptName() { cout << dept; } }; class Student : private Department { public: string name; Student(string n = “Mechanical”, string d=”Electrical”):name(n),Department(d){} using Department::getDeptName; }; </pre>	5+5	CO2

```

int main()
{
Student s (“Civil”);
s.getDeptName();
return 0;
}

```

(b) Enlist at least 5 operator(s) that are not allowed to be overloaded in C++.

OR

Q8. Fill in the blanks to complete the successful execution of the program to read and write text in/from a file.

```

#include <iostream>
#include <fstream>

using namespace std;

int main()
{
    _____; // create object of fstream class

    _____ //opening file “sample.txt” in out (write) mode

    if(!_____ // check for object of fstream class )
    {
        cout<<”Error in creating file!!!”<<endl;
        return 0;
    }

    cout<<”File created successfully.”<<endl;

    _____; //write text into file
    _____; //closing the file

    _____ //again open file in read mode

    if( ! _____ ) //check for object of fstream class
    {
        cout<<”Error in opening file!!!”<<endl;
        return 0;
    }
}

```

CO3

10

	<pre> char ch; //to read single character cout<<"File content: "; while(_____//read until end of file is not found) { _____//read single character from file cout<<ch; } _____//close file return 0; } </pre>		
Q9.	Write a program in C++ to depict the behavior of dominating a data member and over-riding a member function using the concept of “class”. Also, indicate the adequate comment against each expression.	10	CO2, CO3
SECTION-C [40 Marks]			
Q 10.			
(a)	<p>Suggest the expected output of the following program:</p> <pre> #include <iostream> using namespace std; void printChar(int num, char ch); void printChar(char ch , int num); int main() { printChar(5, '@'); printChar('*', 7); return 0; } void printChar(int num, char ch) { int i=0; cout<<endl; for(i=0;i<num;i++) </pre>	5	CO2

	<pre> cout<<ch; } void printChar(char ch, int num) { int i=0; cout<<endl; for(i=0;i<num;i++) cout<<ch; } </pre>		
(b)	<p>Write a program in C++ that demonstrates the process of initialization of Array of Objects (say OBJ1[3] of class type) using the constructor (having 3 arguments to pass). Print the elements of array-object in reverse fashion (of how they are stored in memory).</p>	10	CO1, CO2
(c)	<p>Differentiate between a user defined function and an operator.</p>	5	CO2
Q11.	<p>What error would occur if “cwctype file” is not included in the following program?</p>	2+8	CO3
(a)	<p>Also, suggest the output of the undermentioned code:</p> <pre> #include <cwctype> #include <iostream> using namespace std; int main() { wchar_t rs1 = '.'; wchar_t rs2 = 'c'; wchar_t rs3 = '?'; wchar_t rs4 = 'm'; // Function to check if the character // is a lowercase character or not if (iswlower(rs1)) wcout << rs1 << " is a lowercase "; else wcout << rs1 << " is not a lowercase "; wcout << endl; if (iswlower(rs2)) wcout << rs2 << " is a lowercase "; else wcout << rs2 << " is not a lowercase "; wcout << endl; } </pre>		

	<pre> if (iswlower(rs3)) wcout << rs3 << " is a lowercase "; else wcout << rs3 << " is not a lowercase "; wcout << endl; if (iswlower(rs4)) wcout << rs4 << " is a lowercase "; else wcout << rs4 << " is not a lowercase "; wcout << endl; return 0; } </pre> <p style="text-align: center;">OR</p> <p>With the help of a code (in C++):</p> <ol style="list-style-type: none"> i. Give an example of the situation when we need to open a file for “reading” only. ii. What are file streams and their various types? 	5+5	
(b)	<p>Define:</p> <ol style="list-style-type: none"> i. Abstraction ii. Encapsulation iii. Polymorphism iv. Inheritance v. Function overloading vi. Member access operator vii. return 1; viii. copy constructor ix. friend x. access specifier <p style="text-align: center;">OR</p> <p>How can you identify the application of “Composition-a stronger aggregation” and “Inheritance”?</p> <p>Write a program in C++ to illustrate the difference between the aforementioned terms.</p>	10 10	CO1, CO2, CO3

Name:

Enrolment No:



UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

End Semester Examination, May 2019

Programme Name: B.Tech. ECE
Course Name : OOPs with C++
Course Code : CSEG 2001
Nos. of page(s) : 07
Instructions:

Semester : IV
Time : 03 hrs
Max. Marks : 100

SECTION A [20 Marks]

S. No.		Marks	CO
Q 1.	<p>I. Which of the following operators cannot be opted for overloading?</p> <p>a. == b. :: c. -> d. ()</p> <p>II. What is the syntax of the 'new' and 'delete' operator for allocating/deallocating memory for a single variable and an array?</p>	2+2	CO2
Q 2.	<p>Try to fetch the expected output for the following code:</p> <pre>#include <iostream> using namespace std; class OOPS { int i; public: OOPS (int ii) : i(ii) {} const OOPS operator*(const OOPS & rv) { cout << "Executes *" << endl; return OOPS (i * rv.i); } OOPS & operator+=(const OOPS & rv) { cout << "Executes +=" << endl; i += rv.i;</pre>	4	CO2

	<pre> return *this; } }; int main() { int i = 1, j = 2, k = 3; k += i * j; OOPS ii(1), jj(2), kk(3); kk += ii * jj; return 0; } </pre>		
Q 3.	<p>Consider the following code segment. Assume that the sizeof(int) = 4 and sizeof(double) = 4.</p> <pre> class base { static int statInt; double arr[8]; void display() {} }; class base1: public base { }; class base2: public base { }; class derived: protected base1, protected base2 { }; </pre> <p>What will be the size (in Bytes) of the object derived?</p>	4	CO3
Q 4.	Compare the characteristics of a Constructor and Destructor.	4	CO1, CO2
Q.5	What is Polymorphism? Explain how the different types of Polymorphism can be realized in C++?	4	CO3
SECTION B [40 Marks]			
Q 6. (a)	What are the operators which cannot be overloaded in C++?	5+5	CO2, CO1
(b)	<p>This program is all about the implementation of Pre/Post Decrementer. Fill the blank by keeping this in mind so that the given test cases will satisfy:</p> <pre>#include <iostream></pre>		

```

using namespace std;

class DClass

{ int data;
public:
    _____ {} // Define Constructor

    DClass& operator-- ()
    { --data;
    return _____;
    }

    _____ {
    DClass t(data);
    --data;
    return _____; }

    void disp()
    { cout << " " << data ; }
};

int main()
{
int x;
cin >> x;

DClass obj1(x);
obj1.disp();

DClass obj2 = obj1--;
obj2.disp();

obj2 = --obj1;
obj2.disp();

return 0;
}

```

Q 7. “Can we have more than one generic data type in a class template?” Write a program to support your answer with adequate comment lines. **10** **CO3**

Q8. Write the syntax of the following terms w.r.t. C++:

	<p>(a) Overloaded assignment operator (b) Friend function (c) Constructor Initialization list (d) Multilevel inheritance (e) Pure virtual function</p>	10	CO1, CO2
Q 9.	<p>Design a C++ program to perform user-defined Exception handling. OR With suitable code snippets explain how exceptions can be thrown and re-thrown?</p>	10	CO3
SECTION-C [40 Marks]			
Q 10.			
(a)	<p>What are the different ways in which code reusability can be achieved? With suitable code snippets illustrate the different types of Inheritance. OR Consider the following code. Insert the code for error handling using exception, in editable section to match the test cases.</p> <pre> #include <iostream> #include <exception> using namespace std; class myexception : _____ // Inherit exception with appropriate visibility { virtual const char* what() const throw() { return "DivideByZero"; } }; class DivideByZero { public: int numerator, denominator; DivideByZero(int a = 0, int b = 0) : numerator(a), denominator(b) {} int divide(int numerator, int denominator) { if (denominator == 0) { _____ // Call exception suitably to handle divide by zero error } return (numerator / denominator); } </pre>	3+5	CO2

	<pre>}; int main() { DivideByZero d; int a, b; cin >> a >> b; try { d.divide(a, b); } catch (exception& e) { cout << e.what() << endl; } return 0; }</pre>		
(b)	<p>Define an Abstract Class? Discuss the following cases, regarding the abstract classes, via suitable example in C++:</p> <ol style="list-style-type: none"> i. Having at least 1 pure virtual function ii. User may have pointer(s) and reference(s) of abstract class type iii. Circumstances under which derived class also becomes abstract class <p style="text-align: center;">OR</p> <p>Ambiguity usually occurs during Multipath Inheritance. State the reason of this ambiguity and write a program in C++ to resolve this ambiguity via every possible method.</p>	12	CO3, CO2
Q11.			
(a)	<p>Fill in blank with proper access specifier and function definitions of the class Stack to get the output as per the test cases:</p> <pre>#include <iostream> #include <vector> #include <string.h> using namespace std; class Stack { _____: // Write the appropriate Access specifier vector<char> data_; int top_; public: int empty() { _____; } void push(char x) { _____; } void pop() { _____; } char top() { _____; } }; int main()</pre>	5	CO1, CO3

	<pre> { Stack s; char str[20]; cin >> str; s.data_.resize(100); s.top_ = -1; for(int i = 0; i < strlen(str) ; ++i) s.push(str[i]); while (!s.empty()) { cout << s.top(); s.pop(); s.pop(); } return 0; } </pre>		
(b)	<p>What will be the output of the following code?</p> <pre> #include <iostream> using namespace std; class B { int id; public: static int count; B() { count++; id = count; cout << id << " "; } }; class D : public B { int n; public: D() { count- -; n = count; cout << n << " "; } }; int B::count = 5; </pre>	5	CO1

	<pre> int main() { B *basePtr = new D[2]; delete [] basePtr; return 0; } </pre>		
(c)	<p>Define:</p> <ol style="list-style-type: none"> i. :: ii. Run time polymorphism iii. Object iv. Constructor v. Virtual function vi. exit() vii. Derived data type viii. Dynamic memory allocation ix. Exception Handling x. Return type 	10	CO1, CO2, CO3