Bangladesh Army University of Science and Technology

Department of Computer Science and Engineering

Final Examination, Fall 2018 **Level-2 Term-I**
Course Code: CSE 2105 **Course Title: Object Oriented Programming I**
Time: 03 (Three) hours **Full Marks: 210**

N.B. (i) Answer any three questions from each PART (ii) Use separate answer script for each PART
(iii) Marks allotted are indicated in the margin (iv) User-response in sample execution is underlined

PART A

(Answer any three questions)

1. a) Explain how encapsulation ensure object constructor in using inheritance and polymorphism. 9

b) What is the purpose of constructors and destructors? Write three special properties of a constructor that make it distinct from other member functions. In case of multiple and multilevel inheritance, mention the order in which constructors and destructors are invoked. Give example of each. 4+6 +4 =14

c) What is the problem with the following code segment? How can you solve it? 6+6 =12

```
class myClass
{
      int *ptr;
public:
      myClass(int arg)
      {
            ptr = new int;
            *ptr = arg;
      }

      ~myClass()
      {
            delete ptr;
      }
      int getVal()
      {
            return *ptr;
      }
};
```

```
void printInt(myClass var)
{
      cout<<var.getVal();
}

int main()
{
     myClass obj1(20);
     printInt(obj1);
     cout<<obj1.getVal();
}
```

2 a) What is the purpose of virtual function and why is it important? What are the differences between virtual function and pure virtual function? If a class declaration contains a pure virtual function, what is that class called and what restrictions apply to its usage? 10

b) What is an in-line function? Give an example. What are the advantages and disadvantages of using inline functions? 6

c) Mention three different scenarios in which copy constructor gets called. 6

d) Suppose there are 7 methods defined as following: 8

```
void funl(), void fun2(), void fun3(), void fun4(),
void fun5(), void fun6(), and void fun7()
```

There are also 4 abstract classes named abs1, abs2, abs3, and abs4.

There is also a class named as `MySpecialClass` that needs to be forced to implement all the 7 methods. But there are some constraints

i) Each abstract class can define at most 2 methods.

ii) The class `MySpecialClass` can only inherit 1 class

Write C++ code for `MySpecialClass` to achieve this scenario.

e) Write the difference between an instance member variable and a static member variable. Give an example. 5

3 a) Write all the incorrect statements that will produce compile time error in the following code segments. 15

```
class Base
{
public:
    int m_public;
private:
    int m_private;
protected:
    int m_protected;
 };

class  PRO:  protected Base
{
public:
  PRO()
 {
   m_public = 4;
   m_private = 5;
   m_protected = 6;
 }
};
```

```
class PUB: public Base
{
public:
   PUB()
  {
   m_public = 1;
   m_private = 2;
   m_protected = 3;
  }
};
class PRI: private Base
{
public:
  PRI()
  {
   m_public = 7;
   m_private = 8;
   m_protected = 9;
  }

};
```

```
int main()
{
    Base base;
    base.m_public = 10;
    base.m_private = 11;
    base.m_protected = 12;

    PUB pub;
    pub.m_public = 13;
    pub.m_private = 14;
    pub.m_protected = 15;

    PRO pro;
    pro.m_public = 16;
    pro.m_private = 17;
    pro.m_protected = 18;

    PRI pri;
    pri.m_public = 19;
    pri.m_private = 20;
    pri.m_protected = 21;
}
```

b) What is function overloading? Write some necessary overloaded version of a function called `max(arg1, arg2)` that returns the bigger of the two numeric arguments. The numeric arguments can be char, int, float, and double. 6+4 =10

Also write a single generic function that can achieve the functionality of all the overloaded functions.

c) Why operator overloading utilizes friend function? Demonstrate with a short example. 10

4. a) What is default value in the context of a function parameter? How is it done? Give an example. 6

b) What is late binding? Demonstrate how C++ achieves polymorphism through late binding with appropriate example-code. 14

c) Read the following code segment. What is wrong within the `main()` function? Modify the code segment to solve the issue. 15

```
class A{
 public:
  void eat()
  {
    cout<<"A";
  }
};
```

```
class B: public A{
 public:
  void sleep()
  {
    cout<<"B";
  }
};
```

```
class C: public A{
 public:
  void play()
  {
    cout<<"C";
  }
};
```

```
class D: public B, public C {
  public:
   void beHappy()
   {
      cout<<"D";
   }
};
```

```
int main(){
    D obj;
    obj.eat();
}
```

PART B

(Answer any three questions)

5. a) Consider the following classes. 20

```
class node
{
    int item;
    node *next;
  public:
    node(int arg) {
        item=arg;
        next=NULL;
    }
    void setNext(node *p) {
        next=p;
    }
    node* getNext() {
        return next;
    }

    int getItem() {
        return item;
    }
};
```

```
class stackInterface
{
  protected:
    node* TOP;

  public:
    void push(int item) = 0;
    void pop() = 0;
    int top() = 0;
    int size() = 0;

};
```

Implement a class `Stack` by inheriting from the abstract `stackInterface` class. Your `Stack` class should implement the four functions as follows:

(i) push - add a new item at the top of the stack.

(ii) pop - remove an item from the top of the stack.

(iii) top - return the item which is at the top of the stack.

(iv) size – return the number of items present in the stack.

b) Create a class named `myClass` with a variable named `objectCount` that keeps track the number of objects created for `myClass`. When the total number of objects is greater than 100, reinitialize object Count to 0. 7

c) Write about role of overloading in attaining run-time polymorphism. 8

6. a) Mention some advantages of exception handling over traditional error handling. 7

b) Why is namespace important for large projects? Mention the reason behind discouraging the usage of the '`using namespace`' with a short example. 3

c) Write the output produced by the following code segment: 5

```
int wait_time = 46;
    try
    {
        cout << "Try block entered.\n";
        if (wait_time > 30)
            throw wait_time;
        cout << "Leaving try block.\n";
    }
    catch(int thrown_value)
    {
        cout << "Exception thrown with "
             << "wait_time equal to " << thrown_value << endl;
    }
    catch(…)
    {
        cout << "I catch everything.\n"
    }
    cout << "After catch block." << endl;
```

d) How are static data member and static member function declared for a class? What are the restrictions placed on static member functions? Give an example. 8

e) Write a program that calculates the total sum of an integer array by dividing and allocating the task into two separate threads t1 and t2. 12

7 a) Consider the following class: 25

```
class TwoDimMatrix {
      int R, C;
      int *mat;
  public:
      void input(int r, int c);
      void print();
      void multiply(const TwoDimMatrix &A);
};
```

This class operates on a two dimensional R×C matrix. You need to write the following methods:

i. `Void input(int r, int c)` to dynamically create and take input for an r×c matrix,

ii. `void print()` to display the matrix, and

iii. `void multiply(const TwoDimMatrix &obj)` to multiply the matrix by another matrix- obj.

Besides you need to create your own exception class named `IncompatibleMatrixException` that needs to be thrown when the `multiply()` tries to multiply two incompatible matrices (when C of first matrix is not same as R of second matrix).

Sample usage of the class:

```
TwoDimMatrix a,b;
a.input(3,4);
b.input(4,7);
try {
      a.multiply(b);
}
catch( IncompatibleMatrixException e){
      e.message();
}
```

b) What is a memory leakage? What condition is responsible for leakage? 5

c) When is using a vector beneficial than using a traditional array? Write a code segment that traverse a `vector<int>` named `myVector` to find the smallest integer within the vector. 5

8. a) Consider the following class declaration: 6 + 24 =30

```
class Vector{
        int x, y, z;
    public:
        _ _ _
        _ _ _
};
```

i) Create necessary constructors to support the following two declarations in the main function:

```
Vector p1, p2, p3;
Vector p4 (9, 8, 4);
```

ii) Write member functions and/or friend functions to fulfill the following requirements:

Requirement	Sample Code	Explanation
Addition of two vector objects	p1 = p2 + p3;	Corresponding components of the vectors are added.
Adding an integer to a vector object	p3 = p4 + 100;	Increasing a vector by an integer n implies each component of the vector is to be increased by n.

Increment a vector object	p4++;
	++p4;
		Increasing each component by 1.

Print vector components on screen	cout<<p4;	Prints "point (9, 8, 4)" on the screen.
Distance from origin	int dist = p4;	$dist = \sqrt{9^2 + 8^2 + 4^2}$

b) How does a copy constructor differ from the assignment operator (=)? Give an example. 5