CONFIDENTIAL

Infrastructure
University
Kuala Lumpur

FINAL EXAMINATION
JUNE SEMESTER 2015

BACHELOR OF INFORMATION TECHNOLOGY (HONS)
IN SOFTWARE ENGINEERING
BACHELOR OF INFORMATION TECHNOLOGY (HONS)
IN NETWORK TECHNOLOGY
BACHELOR OF COMPUTER SCIENCE (HONS)

ADVANCED PROGRAMMING
(BTT 112)

(TIME : 3 HOURS)

MATRIC NO. : □□□□□□□□□

IC. / PASSPORT NO. : □□□□□□□□□□□□□□

LECTURER : NORADIBAH ADNAN

GENERAL INSTRUCTIONS

1. This question booklet consists of 8 printed pages including this page.
2. Answer **ALL questions** in the **ANSWER BOOKLET**

CONFIDENTIAL

INSTRUCTIONS: **TIME: 3 *HOURS***

SECTION A (30 MARKS)

There are THREE (3) questions in this section. Answer ALL Questions in the Answer Booklet.

1. Consider the following class prototype of `Square` class where the `side` member is `double`.

```
class Square
{
   public:
      Square(double);         // constructor with a parameter
      void setSide(double); // to initialize a member variable
      double getSide(); // to return a value of member variable
      double area();     // formula area = side x side
   private:
      double side;       // member variable declaration
};
```

Answer the following:

a) Rewrite the class `Square` to include an empty exception class. The exception class is to ensure no negative value shall be accepted during the execution time.
(2 marks)

b) Write a function definition for `setSide()` function, to throw an exception class if value entered by user is a negative.
(6 marks)

c) If the `Square` class has a copy constructor prototype as following:

`Square(const Square &);`

Write its copy constructor definition.
(4 marks)

2. Convert the following by using template function that can accept any data type:

```
char showValue(char value[], int sizeValue)
{
    cout << "Value are: ";
    for(int x = 0; x < sizeValue; x++)
        cout << value[x] << " ";
}
```

(4 marks)

3. Consider the following class that using the operator overloading for `operator >`

```
class OpOverLoad
{
   public:
      OpOverLoad() { this -> number = 1.0; }
      OpOverLoad(double number) { this -> number = number;   }
      double getNumber() { return number; }
      friend bool operator > (OpOverLoad);

   private:
      double number;
};
```

Answer the following:

a) Write the definition for `operator >`.

(9 marks)

b) Demonstrate the overloaded `operator >` in the `main()` function.

(5 marks)

SECTION B **(20 MARKS)**

There are FIVE (5) questions in this section. Answer ALL Questions in the Answer Booklet.

1. Consider the following class:

```
class BaseClass
{
   protected: int x;
   public:
            void print() const
            { cout << "x = " << x << endl;   }
            int getX()const { return x; }
            BaseClass(int a = 0) { x = a; }
};
class DerivedClass  : public BaseClass
{
   private: int y;
   public:
            void print()const
            { cout << "x = " << x << ", y = " << y << endl
                   << "x + y = " << x + y << endl; }
            int getResult() const { return x + y; }
            DerivedClass(int a = 0, int b = 0) ;
};
DerivedClass::DerivedClass(int a, int b)  : BaseClass(a)
{ y = b ; }
```

a) What is the output for the following code:

```
BaseClass base(7);
base.print();
cout << "=== " << base.getX() << endl;
```

(2 marks)

b) What is the output for the following code:

```
DerivedClass derived(3, 8);
derived.print();
cout << "### " <<  derived.getX() << endl;
```

(4 marks)

2. What is the output of the following:

```
class Calculate
{
    private:
        int x;
        static int y;
    public:
            static int countNum;
            void print() const
            {
                cout << "x = " << x << ", y = " << y
                     << ", count = " << countNum << endl;
            }

            void setX(int a) { x = a; }
            static void incrementY() { y++; }
            Calculate(int a = 0) { x = a; }
};
int Calculate::countNum = 0;
int Calculate::y = 0;

int main()
{
    Calculate c1(3), c2(5);

    Calculate::incrementY();
    Calculate::countNum++;
    c1.print();
    c2.print();

    cout << endl;
    c1.incrementY();
    c1.setX(8);
    c1.print();
    c2.print();

    cout << endl;
    c2.setX(23);
    c2.print();

    return 0;
}
```

(5 marks)

3. What is the output by the following program segment?

```
char A[] = "ABCDE";
char *ptr = A;
++ptr;
cout << ptr << endl;
ptr += 2;
cout << ptr << endl;
--ptr;
cout << ptr << endl;
```

(3 marks)

4. What is the output by the following program segment?

```
int main()
{
    ofstream outputFile;
    outputFile.open("File1.txt");

    cout <<"Writing data to the file.\n";
    outputFile << "One\n" << "Two\n" << "Three";
    outputFile.close();
    cout <<"Complete writing on file.\n";

    return 0;
}
```

(3 marks)

5. What is the output by the following program segment?

```
int main()
{
    double num = 17.816392;
    fstream dataFile("File1.txt", ios::out);

    dataFile << fixed << num << endl;
    dataFile << setprecision(3) << num << endl;
    dataFile << setprecision(1) << num << endl;

    return 0;
}
```

(3 marks)

SECTION C (50 MARKS)

There is ONE (1) question in this section. Answer the question in the Answer Booklet.

1. Most teachers at the Excel International school assign various graded activities for their students to complete. A graded activity can receive a numeric score such as 70, 85, 90 and so on, and a letter grade such as A,B, C, D or F. The following class is designed to hold the numeric score and letter grade of a graded activity. When a numeric score is stored by the class, it automatically determines the letter grade.

   Given is inline class definition for class `GradedActivity`:

```
class GradedActivity
{
      protected:
            double score;
      public:
      GradedActivity(){ score = 0.0; }
      GradedActivity(double s){ score = s; }
      void setScore(double s){ score = s; }
      double getScore() const { return score; }
      virtual char getLetterGrade() const;
};
```

   Write a class **prototype** and **definition** for `PassFailActivity` that inherits from class `GradedActivity`. The class is intended to determine a letter grade of 'P' for passing, or 'F' for failing. The `PassFailActivity` class has following member variable and functions:

   - A member variable to store the minimum passing score for an activity. (9 marks)

   - A default constructor. A default value for minimum passing score is 0.0. (3 marks)

   - Constructor with one parameter (5 marks)

   - A mutator (set) function, to validate the input. (5 marks)

   - An accessor (get) function (2 marks)

   - A redefine function of `getLetterGrade()`from class `GradedActivity`. This function returns a grade of 'P' if the numeric score is greater than or equal to minimum passing score. Otherwise the function returns a grade of 'F'. (8 marks)

Write a `main()` function to demonstrate the concept of polymorphism for `GradedActivity` and `PassFailActivity` classes.

(9 marks)

Write a function definition to displays a numeric score and letter grade. The function uses a base class pointer as its parameter.

(9 marks)

*** **END OF QUESTIONS** ***