**Spring 2021 FRE-GY 6883 Financial Computing 1st Quiz (Saturday Afternoon)**

Last Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_First Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_EMAILID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Complete the implementation of the constructor with parameters for class ByTrapezoid and BySimpson.**

**You are not allowed to create other member functions.**

1. **Complete the main() function to get the expected results.**

**You must use cal1 and cal2 for your calculation. You cannot use ptr1 or ptr2 directly for your calculation.**

|  |
| --- |
| #include <iostream>  using namespace std;  class DefInt  {  protected:  double a, b;  int N;  double (\*f)(double x);  public:  DefInt(double a\_, double b\_, int N\_, double (\*f\_)(double x))  {  a = a\_; b = b\_; N = N\_, f = f\_;  }  virtual double Approximation() = 0;  virtual ~DefInt() {}  };  class ByTrapezoid : public DefInt  {  public:  // Complete the implementation of the constructor with parameters for class ByTrapezoid.  // You are not allowed to create other member functions.  ByTrapezoid(double a\_, double b\_, int N\_, double(\*f\_)(double x)) :DefInt(a\_, b\_, N\_, f\_) {}  double Approximation();  ~ByTrapezoid() {}  };  double ByTrapezoid::Approximation()  {  cout << "ByTrapezoid: ";  double h = (b - a) / N;  double Result = 0.5 \* f(a);  for (int n = 1; n < N; n++) Result += f(a + n \* h);  Result += 0.5 \* f(b);  return Result \* h;  }  class BySimpson : public DefInt  {  public:  // Complete the implementation of the constructor with parameters for class BySimpson.  // You are not allowed to create other member functions.  BySimpson(double a\_, double b\_, int N\_, double(\*f\_)(double x)) :DefInt(a\_, b\_, N\_, f\_) {}  double Approximation();  ~BySimpson() {}  };  double BySimpson::Approximation()  {  cout << "BySimpson ";  double h = (b - a) / N;  double Result = f(a);  for (int n = 1; n < N; n++) Result += 4 \* f(a + n \* h - 0.5 \* h) + 2 \* f(a + n \* h);  Result += 4 \* f(b - 0.5 \* h) + f(b);  return Result \* h / 6;  }  class Calculation  {  private:  DefInt\* ptr;  public:  void SetPtr(DefInt\* ptr\_)  {  ptr = ptr\_;  }  double Appximation()  {  return ptr->Approximation();  }  };  double f(double x) { return x \* x \* x - x \* x + 1; }  int main()  {  double a = 1.0;  double b = 2.0;  int N = 1000;  DefInt\* ptr1 = new ByTrapezoid(a, b, N, f);  DefInt\* ptr2 = new BySimpson(a, b, N, f);  Calculation cal1, cal2;  // Complete the missing codes in main function. You must use cal1 and cal2 for your calculation.  // You cannot use ptr1 or ptr2 directly for your calculation  cal1.SetPtr(ptr1);  cal2.SetPtr(ptr2);  cout << cal1.Appximation() << endl;  cout << cal2.Appximation() << endl;  delete ptr1, ptr2;  return 0;  }  /\*  ByTrapezoid: 2.41667  BySimpson 2.41667  \*/ |