שאלה 3: C++ (35 נקודות) – מועד ב' קיץ 2002

סעיף א' (18 נקודות)

```
#ifndef QUEUE_H_
#define QUEUE_H_

#include <iostream>
using std::ostream;
using std::istream;

template<class T, int Dim>
class queue {
  T arr[Dim];
  int size, first;

public:
  queue();
  queue(const T&);

  queue &enqueue(const T&);
  queue &append(const queue &);
  queue &transfer(queue &);
  queue &clear();
  bool empty() const;
  T dequeue();
  T first() const;
};

template<class T, int Dim>
ostream &operator<<(ostream &,
                    const queue<T,Dim> &);

template<class T, int Dim>
istream &operator>>(istream &, queue<T,Dim> &);

template<class T, int Dim>
queue<T,Dim> operator+(const queue<T,Dim> &,
                       const queue<T,Dim> &);

#endif
```

מימוש המזכה בבונוס:

סעיף ב' (12 נקודות)

```
class QueueBaseSize {
protected:
  int maxsize, size;
public:
  explicit QueueBaseSize(int maxsize_);
  bool empty() const;
};

template<class T>
class QueueBase : public QueueBaseSize {
  T *arr;
  int first;
public:
  explicit QueueBase(int size);
  QueueBase(const QueueBase &);
  ~QueueBase();

  QueueBase &operator=(const QueueBase &);

  QueueBase &enqueue(const T&);
  QueueBase &append(const QueueBase &);
  QueueBase &transfer(QueueBase &);
  QueueBase &clear();
  T dequeue();
  T first() const;
  void print(ostream &) const;
  void read(istream &);
  QueueBase operator+(const QueueBase &);
};

template<class T>
ostream &operator<<(ostream &,
                    const QueueBase<T> &);
template<class T>
istream &operator>>(istream &, QueueBase<T> &);

template<class T, int Dim>
class queue : QueueBase<T> {
public:
  queue();
  queue(const T &elem);
};
template<class T, int Dim>
queue<T,Dim> operator+(const queue<T,Dim> &,
                       const queue<T,Dim> &);
```

```
template<class T, int Dim>
queue<T,Dim>::queue() : QueueBase<T>(Dim) {}

template<class T, int Dim>
queue<T,Dim>::queue(const T &elem)
    : QueueBase<T>(Dim)
{
  arr[0] = elem;
  size++;
}


template<class T>
QueueBase<T> & QueueBase<T>::enqueue(const T
&e)
{
  if(size>=maxsize) exit(1);
  arr[(first+size) % maxsize] = e;
  size++;
}


template<class T>
QueueBase<T> QueueBase<T>::
        operator+(const QueueBase<T> &q)
{
  if(size+q.size>maxsize) exit(1);

  QueueBase<T> tmp = *this;

  int i = q.first, num = q.size;
  while(num>0) {
    tmp.arr[(tmp.first+tmp.size)%maxsize] = q.arr[i];
    tmp.size++;

    i = (i+1)%q.maxsize;
    num--;
  }
  return tmp;
}
```

שאלה 3: C++ (35 נקודות) - מועד א' חורף 2002-2003

סעיף א' (18 נקודות)

```
#ifndef POINT_H_
#define POINT_H_

template<class Coordinate>
class Point
{
  Coordinate x, y;

public:
  Point(const Coordinate &, const Coordinate &);

  double  radius() const;
  double  distance(const Point &) const;
  Point    middle(const Point &) const;
  double  arg(); const;

  Point operator+(const Point &) const;
  Point &operator+=(const Point &);

  Point operator*(const double &) const;
  Point &operator*=(const double &);
};

template<class Coordinate>
Point<Coordinate> operator*(const double &,
                            const Point<Coordinate> &);

#endif
```

סעיף ב' (12 נקודות)

```
template<class Coordinate>
Point<Coordinate>::Point(const Coordinate &xval,
                         const Coordinate &yval)
    : x(xval), y(yval) {}

template<class Coordinate>
Point<Coordinate> Point<Coordinate>::operator+
                (const Point<Coordinate> &p) const
{
  return Point<Coordinate>(x+p.x, y+p.y);
}

template<class Coordinate>
Point<Coordinate> &Point<Coordinate>::operator+=
                        (const Point<Coordinate> &p)
{
  x += p.x;
  y += p.y;
  return *this;
}


#include <cmath>
template<class Coordinate>
double Point<Coordinate>::distance
                (const Point<Coordinate> &p) const
{
  return sqrt((x-p.x)*(x-p.x)+(y-p.y)*(y-p.y));
}
```

סעיף ג' (5 נקודות)

```
template<class Coordinate>
class PointBase
{
  Coordinate *coords;

public:
  PointBase(int dim, const Coordinate init[]);
  ~PointBase();

  PointBase middle(const PointBase &) const;
  PointBase operator+(const PointBase &) const;
};


template<class Coordinate, int Dim>
class Point : public PointBase<Coordinate>
{

public:
  Point(const Coordinate init[]) :
              PointBase<Coordinate>(Dim,init) {}
};
```

סעיף ג' (5 נקודות) (מועד ב' קיץ 2002)

```
class queue {
public:
  class Exception {}
  class Underflow : public Exception {}
  class Overflow : public Exception {}
};

queue &queue::enqueue(const T &)
{
   …
   throw queue::Overflow();
   …
}

queue &queue::dequeue(const T &)
{
   …
   throw queue::Underflow();
   …
}

int main()
{
  …
  try {
    …enqueue(…);
    …dequeue(…);
  }
  catch(const queue::Overflow &over) {
    …
  }
  catch(const queue::Underflow &under) {
    …
  }
  …
}
```