חורף שאלה 1 - ADT

א. פתרון

```
typedef struct AdptArray_
{
	int ArrSize;
	PElement* pElemArr;
	DEL_FUNC delFunc;
	COPY_FUNC copyFunc;
}AdptArray;
```

ב. פתרון

```
PAdptArray CreateAdptArray(COPY_FUNC copyFunc_, DEL_FUNC delFunc_)
{
	PAdptArray pArr = (PAdptArray)malloc(sizeof(AdptArray));
	if (pArr == NULL)
		return NULL;
	pArr->ArrSize = 0;
	pArr->pElemArr = NULL;
	pArr->delFunc = delFunc_;
	pArr->copyFunc = copyFunc_;
	return pArr;
}
Result SetAdptArrayAt(PAdptArray pArr, int idx, PElement pNewElem)
{
	PElement* newpElemArr;
	if (pArr == NULL)
		return FAIL;

	if (idx >= pArr->ArrSize)
	{
```

```
// Extend Array
		if ((newpElemArr = (PElement*)calloc((idx + 1), sizeof(PElement))) ==
NULL)
			return FAIL;
		memcpy(newpElemArr, pArr->pElemArr, (pArr->ArrSize) *
sizeof(PElement));
		free(pArr->pElemArr);
		pArr->pElemArr = newpElemArr;
	}

	// Delete Previous Elem
	pArr->delFunc((pArr->pElemArr)[idx]);
	(pArr->pElemArr)[idx] = pArr->copyFunc(pNewElem);

	// Update Array Size
	pArr->ArrSize = (idx >= pArr->ArrSize) ? (idx + 1) : pArr->ArrSize;
	return SUCCESS;
}
```

ג. פתרון

```
void DeleteAdptArray(PAdptArray pArr)
{
	int i;
	if (pArr == NULL)
		return;
	for(i = 0; i < pArr->ArrSize; ++i)
	{
		pArr->delFunc((pArr->pElemArr)[i]);
	}
	free(pArr->pElemArr);
	free(pArr);
```

```
}
```

ד. פתרון

```
PElement GenTreeChildArrCopyFunc(PElement pElem);
void GenTreeChildArrDeleteFunc(PElement pElem);


PGenTree CreateGenTree(PElement pElem_, COPY_FUNC copyFunc_, DEL_FUNC
delFunc_, PRINT_FUNC printFunc_)
{
	PGenTree pRoot = (PGenTree)malloc(sizeof(GenTree));
	if (pRoot == NULL)
		return NULL;
	pRoot->fatherChildIdx = -1;
	pRoot->pElem = copyFunc_(pElem_);
	pRoot->pFather = NULL;
	pRoot->pChildrenArr = CreateAdptArray(GenTreeChildArrCopyFunc,
GenTreeChildArrDeleteFunc);
	pRoot->delFunc = delFunc_;
	pRoot->copyFunc = copyFunc_;
	pRoot->printFunc = printFunc_;
	return pRoot;
}
```

ה. פתרון

```
Result SetChildTreeAt(PGenTree pRoot, int childIdx, PGenTree pChild)
{
	Result res;
	// Check if Tree or Subtree are Initialized
	if (pRoot == NULL || pChild == NULL || pChild->pFather != NULL)
		return FAIL;


	// Connect Trees
```

```
	res = SetAdptArrayAt(pRoot->pChildrenArr, childIdx, (PElement)pChild);
	if (res == SUCCESS)
	{
		pChild->pFather = pRoot;
		pChild->fatherChildIdx = childIdx;
	}
	return res;
}

// Implement Internal Functions
PElement GenTreeChildArrCopyFunc(PElement pElem)
{
	return pElem;
}



void GenTreeChildArrDeleteFunc(PElement pElem)
{
	DeleteGenTree((PGenTree)pElem);
}
```

ו. פתרון

```
void DeleteGenTree(PGenTree pRoot)
{
	PGenTree pFather;
	// check if tree exists
	if (pRoot == NULL)
		return;
	// Detach tree from father node (also deletes)
	if (pRoot->pFather != NULL)
```

```
        {
                pFather = pRoot->pFather;
                pRoot->pFather = NULL;
                SetAdptArrayAt(pFather->pChildrenArr, pRoot->fatherChildIdx,
    NULL);
        }
        else // This is a true root
        {
                DeleteAdptArray(pRoot->pChildrenArr);
                pRoot->delFunc(pRoot->pElem);
                free(pRoot);
        }
    }
```

שאלה 2

א) הגדרות המחלקות:

```
class Vertex
{
public:
    Vertex(int id, const char* tag = "");
    ~Vertex();

    double GetOutput();
    void ConnectToVertex(Vertex* vertex, double weight);
    virtual void Print(std::ostream& output) const;
    virtual void Simulate() = 0;

protected:
    std::vector<Vertex*> m_connectedVertices;
    std::vector<double> m_weights;

    int m_id;
    double m_output;
    char* m_tag;
};

class InputVertex :
        public Vertex
{
public:
        InputVertex(int id, const char* tag);
        ~InputVertex();
```

```
        void PushInput(double value);
        virtual void Simulate();
        void Print(std::ostream & ro) const;

protected:
        std::queue<double> m_inputs;
};

class ReLUVertex :
        public Vertex
{
public:
        ReLUVertex(int id, const char* tag = "", double bias = 0);
        ~ReLUVertex();

        virtual void Simulate();
        virtual void Print(std::ostream& ro) const;

protected:
        double m_bias;
};
```

ב) מימוש Simulate:

ב-Vertex אין צורך (וירטואלי טהור).

```
void InputVertex::Simulate()
{
        double value;
        if (!m_inputs.empty())
        {
                value = m_inputs.front();
                m_inputs.pop();
        }
        else
                throw "No inputs in queue!";

        m_output = value;
}

void ReLUVertex::Simulate()
{
        double sum = 0;

        if (m_connectedVertices.size() == 0)
                throw "No Input!";

        for (int i = 0; i < m_connectedVertices.size(); i++)
                sum += m_connectedVertices[i]->GetOutput() * m_weights[i];
        m_output = std::max(0.0, sum + m_bias);
}
```

ג) אובייקט שמעוניין לתפקד כמידע חייב לממש את האופרטורים:

Object::operator+(Object) // for sums
Object::operator*(double) // for multiplication by edge weight
operator<<(ostream, object) // for printing

שאלה 3 – C++ (20 נק')

1) חייבים להיות:
   a. קונסטרקטור דיפולטי
   b. קונסטרקטור העתקה
   c. אופרטור השמה
   d. דיסטרקטור

```
char* createNewCopy(const char* src) {
        char* dst;
        if (src==NULL) return NULL;
        dst = new char[strlen(src)+1];
        strcpy(dst,src);
        return dst;
}

class Car {
public:
         Car();
         Car(int LicensePlate, const char* Manufacturer);
         Car(const Car& rhs);
         Car& operator=(const Car& rhs);
         ~Car();
 private:
        int LicensePlate_ ;
         char* Manufacturer_;
};

Car::Car() : LicensePlate_(0) , Manufacturer_(NULL) {}

Car::Car(int LicensePlate, const char* Manufacturer) : LicensePlate_(LicensePlate),
Manufacturer_(createNewCopy(Manufacturer)) {}

Car::~Car() {
 if(Manufacturer_){
  delete [] Manufacturer_;
 }
}

Car& Car::operator=(const Car& rhs){
  if (this != &rhs) {
   LicensePlate_ =rhs.LicensePlate_;
   if (Manufacturer_) {
        delete [] Manufacturer_;
   }
    Manufacturer_ = createNewCopy(rhs.Manufacturer_);
 }
```

```
  return *this;
}

Car::Car(const Car& rhs) :
  LicensePlate_(rhs.LicensePlate_), Manufacturer_(createNewCopy(rhs.Manufacturer_)) {}
```

2) יש להוסיף לכל אחת מהמחלקות מתודה וירטואלית של הדפסה ובנוסף להגדיר אופרטור << חיצוני למחלקות(לא חייב להיות friend) שמקבל פרמטר ostream& ו&const event.

```
char* createNewCopy(const char* src) {
        char* dst;
        if (src==NULL) return NULL;
        dst = new char[strlen(src)+1];
        strcpy(dst,src);
        return dst;
}

class Event {
public:
        Event(const char * Title, const Date& StartDate);
        virtual void print(ostream& os) const;
private:
        const char * Title_;
        Date StartDate_;
};

Event::Event(const char * Title, const Date& StartDate):
 Title_(createNewCopy(Title)),StartDate_(StartDate) {}

void Event::print(ostream& os) const {
        os<<Title_<<" is at
        "<<StartDate_.theDay()<<"/"<<StartDate_.theMonth()<<"/"<<StartDate_.th
        eYear()<<endl;
}

class Meeting : public Event {
public:
        Meeting(const char * Title, const Date& StartDate, const char * Location,
const char* WithWho);
        virtual void print(ostream& os) const;
private:
        char* Location_;
        char* WithWho_;
 };
```

```
Meeting::Meeting(const char * Title, const Date& StartDate, const char * Location,
const char* WithWho) :

Event(Title,StartDate),Location_(createNewCopy(Location)),WithWho_(createNewC
opy(WithWho)) {}

void Meeting::print(ostream& os) const {
        os<<"Meeting: ";
        Event::print(os);
         os<<"Location: "<<Location_<<endl;
         os<<"With: "<<WithWho_<<endl;
}

class Vacation : public Event {
public:
        Vacation(const char * Title, const Date& StartDate, const char * Location,int
numDays);
        virtual void print(ostream& os) const;
private:
        char* Location_;
        int NumDays_;
};

Vacation::Vacation(const char * Title, const Date& StartDate, const char *
Location,int NumDays) :
 Event(Title,StartDate),Location_(createNewCopy(Location)),NumDays_(NumDays) {}

void Vacation::print(ostream& os) const {
          os<<"Vacation: ";
         Event::print(os);
         os<<"Location: "<<Location_<<endl;
        os<<"Num of days: "<<NumDays_<<endl;
}

ostream& operator<<(ostream& ro, const Event& event){
        event.print(ro);
        return ro;
}
```

שאלה 4 BASH

סעיף א':

average:

```
#!/bin/bash
cat $1 | grep -v "num job" | calc_average
```

calc_average:

```
#! /bin/bash
(( sum=0 ))
(( count=0 ))
while read line; do
line_split=($line)
(( count+=1 ))
(( sum+=${line_split[3]}+0 ))
done
(( avg=$sum/$count ))
echo "$avg"
```

סעיף ב':

average_per_job:

```
#!/bin/bash
jobs=($@)
(( jobs_num=$#-1 ))
for job in ${jobs[@]:1:$jobs_num}; do
avg=`grep $job "$1" | calc_average`
echo "Average salary of $job: $avg"
done
```