National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Object Oriented Programming**	**Course Code:**	**CS217**
Program:	**BS(Computer Science)**	**Semester:**	**Fall 2019**
Duration:	**2.5 hrs (150 min.)**	**Total Marks:**	**80**
Paper Date:	**12-Dec-2019**	**Weight**	**40**
Section:	**All**	**Page(s):**	**10**
Exam:	**Final**	**Roll No:**	

1. Attempt all questions in the space provided in this sheet. You can use **rough sheets** but don't need to attach it here as it will not be marked.
2. Questions during exam are not allowed. Take reasonable assumptions where needed.

Question 01: Give a short answer to the question provided below.
[Marks: 24+6]

I. **a).** Which problem arises due to multiple inheritance, if hierarchical inheritance is used previously for its base classes?
b). How to overcome it?

a). diamond problem
b). by adding the keyword virtual with class (while inheriting)

II. If class A inherits class B and class C as

```
class A: public class B, public class C {
// class body;
};
```

which class constructor will be called first?

B

III. Difference between automatic and static variables

Automatic :

Declared (memory allocated)
Deleted automatically when scope ends

Static:

Declared (memory allocated)
Deleted automatically when program ends

IV. Nowadays every Person has a cell phone. Is it an aggregation or composition? Justify your answer.

aggregation

V. What is down-casting?

Converting base class pointer (or reference) to derived class pointer.

VI. When is copy constructor called?

1. Object is declared from another object
2. Object is passed by value in function
3. When object is returned from function.

VII. Which is the condition that must be followed if the array of objects is declared without initialization, only with size of array?

Default constructor

VIII. Hiding the internal working of a washing machine is an example of Encapsulation or Abstraction.

abstraction

IX. Identify the following code as deep copy, shallow copy, memory leak and dangling pointer.

```
int * p = new int [5];
for (int i =0;i<5;i++)
        p[i]=i;
int * q=new int [5];
for (int i =0;i<5;i++)
        *(q+i)=p[i];
delete [] p;
delete [] q;
```

Answer: **deep copy**

```
int * p = new int[10];
int * q=p;
for (int i =0;i<10;i++)
        p[i]=i;
delete [] p;
for (int i =0;i<10;i++)
        *(q+i)=i*2;
delete [] q;
```

Answer**: Shallow copy & dangling pointer**

```
int * p = new int[10];

for (int i =0;i<10;i++)
        p[i]=i;
p = new int[5];
for (int i =0;i<5;i++)
        *(p+i)=i*2;
delete [] p;
```

Answer**: memory leak**

Question 02: [Marks: 6]

Write a template function ***Insert*** which takes two 1D arrays (***Source*** and ***destination***) along with an index ***i*** . Your function should insert ***destination*** array into ***Source*** array between index i & i+1

For example:

Source:

1	0	1	1	0	0
0	1	2	3	4	5

Destination:

0	0	1
0	1	2

If i=2

Then source will become

Source:

1	0	1	***0***	***0***	***1***	1	0	0
0	1	2	3	4	5	6	7	8

Solution:

```
void Insert(T * & x, T *y, int i, int
X, int Y) {
	T * temp = new T[X + Y];
	int j = 0;
	for ( j = 0; j <= i; j++)
		temp[j] = x[j];
	int k = 0;
	for ( k = 0; k < Y; k++)
		temp[j + k] = y[k];
	int m = j + k;
	for (; j < X; j++) {
		temp[m] = x[j];
		m++;
	}
	delete[]x;
	x = temp;
}
```


Question 03:

[Marks: 15]

Consider the class **Product** that has following members: Product ID, expiry date, cost and name. Write a function ***Sort()*** that takes ***an array of Products*** and its ***size*** as parameters. The function Sort must sort all the products according to their ***expiry date***. In sorted order a product comes first whose expiry date is earlier. If two products are expired on same date then one must come first which is more expansive than the other.

Note that all data members of all the classes are private and there are no friend functions or classes. ***Also, you can't add new data members in the classes.***

For example, if we have following Products

Product Id: 5	Product Id: 8	Product Id: 2	Product Id: 5	Product Id: 5
Expiry date: 3/5/2021	Expiry date: 20/12/2019	Expiry date: 20/12/2019	Expiry date: 3/5/2022	Expiry date: 3/5/2020
Cost: Rs. 200	Cost: Rs. 500	Cost: Rs. 250	Cost: Rs. 340	Cost: Rs. 200
Name: cocopowder	Name: mini pizza	Name: cupcake	Name: chocolate	Name: cupcake

After sorting it will be

Product Id: 8	Product Id: 2	Product Id: 5	Product Id: 5	Product Id: 5
Expiry date: 20/12/2019	Expiry date: 20/12/2019	Expiry date: 3/5/2020	Expiry date: 3/5/2021	Expiry date: 3/5/2022
Cost: Rs. 500	Cost: Rs. 250	Cost: Rs. 200	Cost: Rs. 200	Cost: Rs. 340
Name: mini	Name:	Name:	Name:	Name:

pizza	cupcake	cupcake	cocopowder	chocolate

Solution:

```
class Product{
int productID;
date Expir
};
```

Question 04: Write output of the following code in the space provided.
[Marks: 13]

```
#include <iostream>

#include <conio.h>

using namespace std;

class A {

      friend ostream & operator <<(ostream &, const A & b);

      int m;


public:

      A(int a = 0);

      A operator *(int x);

      A & operator =(const A & a);
```

```
      int get() {
            return m;
      }
      void set(int k) {
            m = k;
      }
};
class B:public A {
      int n;
      friend ostream & operator <<(ostream &, B & b);
public:
      B(int a = 0, int b = 0) :A(a) {
            n = b;
      }
      B & operator =( B & a);
      B operator *(int x);
}
A & A::operator =(const A & a) {
      if (this != &a) {
            m = a.m;
      }
      return *this;
}
B & B::operator =( B & a) {
      if (this != &a) {
            n = a.n;
            set(a.get());
      }
```

```
	return *this;
}
ostream & operator <<(ostream & obj, const A & b) {
	obj << "(" << b.m <<  ")     " ;
	return obj;
}
ostream & operator <<(ostream & obj,  B & b) {
	obj << "(" << b.get() << "," << b.n << ")" << endl;
	return obj;
}
A::A(int a) {
	m = a;
}
A A::operator*(int x) {
	A obj;
	obj.m = m * x;
	return obj;
}
B B::operator*(int x) {
	n =  n * x;
	return *this;
}
A * function(B* arr, int& size)
{
	B *a = arr;
	arr = new B[size];
	for (int i = 0; i < size; i++)
		a[i] = a[i] * 3;
```

```
      B * temp = new B(4, 5);
      *temp = *temp * 2;
      return temp;
}

void main() {
      int size = 5;
      B * p = new B[5]{ B(1, 1), B(2, 2), B(3, 3), B(4, 4), B(5, 5) };
      cout << "p:" << endl;
      for (int i = 0; i < size; i++)
            cout << p[i] ;
      cout << endl;
      A *x = function(p, size);
      cout << "Value of p after function call:" << endl;
      for (int i = 0; i < size; i++)
            cout << p[i] ;
      cout << endl;
      cout << "X: " << *x<<endl;
      cout << "x*3:    " << (*x) * 3 << endl;
      cout << "X: " << *x;

      delete[] p;
      delete[] x;



      _getch();
                                    }
```

Write your outputs below:

p:

Value of p after function call:

X:

X*3:

X:

Question 05: [Marks: 16]

Zain and Umair are playing a game. In this game, there are initially N bins in a circle, conveniently labeled 1 through N in clockwise order — for each valid i, bin i+1 is the next bin clockwise from bin i, and bin 1 is the next bin clockwise from bin N. Also, there is a ball in bin 1.

The game is played with fixed integer parameters R and K. Initially, the players designate one of the bins as a ***special bin***. Then, they alternate turns; Zain plays first. The turns are numbered starting from 1. In each turn, the following happens:
The current player must choose an integer s between 1 and K (inclusive) and move the ball s steps clockwise, i.e. move it to the next bin clockwise (that has not been removed yet) s times.
Then, if the number of the current turn is not divisible by R, nothing happens. Otherwise, let's denote the bin that currently contains the ball by b. The current player must remove bin b from the circle.
If the removed bin is the special bin, the current player wins the game.
Otherwise (only if the number of the current turn is divisible by R), the ball is placed in the next bin clockwise from bin b that has not been removed yet.
It is clear that the game is finite and always has a winner. For each choice of the special bin, determine the winner of the game.

Input:
The input contains three space-separated integers $N, K \wedge R$. Constraints for the problem are given below.

- $1 \le K \le N \le 1000$
- $1 \le R \le 10{,}000$
- **R** is odd

For example, if the sample input is :
5 1 1

Output:
Print a single line containing a string with length N. For each valid i, the i-th character of this string should be 'Z' if Zain wins or 'U' if Umair wins when bin i is special.
For input 5 1 1 the output is:
ZZZUU

```
#include <array>

#include <string>

#include<conio.h>

#include <iostream>
```

```
#include <vector>
using namespace std;


void main() {


	const int N = 10;
	vector<int> v(N,0);// Assign vector
	int R = 0;
	int K = 0;
	int special = 5;
	int s = 1;
	int i = 0;
	int turn = 1;
	int b = 1;
	v[0] = b;
	cout << "Enter R such that            1=R =10,000      " << endl;
	cin >> R;
	cout << "Enter K such that      1=K =N =1000    " << endl;
	cin >> K;



	bool over = false;
	while (!over) {
		cout << "Enter a number between 1 and " << R << endl;
		cin >> s;
		s--;
		v[i] = 0;
		i = (i + s) % N;
		if (i == special) {
			over = true;
```

```
                    if (turn % 2 == 0)
                          cout << "Z";
                    else
                          cout << "U";
             }
             else
             {
                    v[i] = b;
                    if (turn %R == 0) {
                          int x = (i + 1) % N;
                          v[x] = b;
                          vector<int>::iterator p;
                          p = v.begin();
                          p = p + i;
                          v.erase(p);
                          i = x;

                    }
                    turn++;

             }
      }


      _getch();

}
```

Rough Work