National University of Computer and Emerging Sciences, Lahore Campus

Course:	Computer Programming	Course Code:	CS103
Program:	BS(Computer Science)	Semester:	Fall 2018
Duration:	45 Minutes	Total Marks:	30
Paper Date:	21-Dec-2018	Weight	10
Section:	All	Page(s):	3
Exam:	Final – Part I	Roll No: Section	

Instructions:
- Attempt Part I in the space provided in this sheet.
- Questions during exam are not allowed. Take reasonable assumptions where needed.

Question [10x3 = 30 Marks] For the code segments given below, determine the output/error(s). If there is any error, highlight the exact line that will cause the error.

Part(i)

```
class Date{
private:
	static int day;
	static int month;
	static int year;
public:
	Date(int d=1, int m =1 , int y = 2000){
		day = d; month = m; year = y;
	}
	void Print(){
	  cout<<day<<"-"<<month<<"-"<<year<<endl;
	}
};//end of Date class
class Student{
private:
	char name[20];
	Date dateOfBirth;
	static int* marks;
public:
	Student(char* n = "", int d=1, int m=1,
int y=2000, int* _marks = 0):dateOfBirth(d,m,y){
		strcpy(name, n);
		if(_marks != 0){
			marks = new int[5];
			for(int i=0 ; i<5; i++){
				marks[i] = _marks[i];
			}
		}
		else
			marks = 0;
	}
	void Print()
	{
		cout<<name<<"\t";
		dateOfBirth.Print();
		for(int i=0 ; i<5; i++){
		     cout<<marks[i]<<" ";
		}
		cout<<endl;
	}
	~Student(){
		if(marks != 0)
			delete[] marks;
	}
};//end of student class
int Date::day = 1;
int Date::month = 1;
int Date::year = 2000;
int* Student::marks = 0;
void main()
{
  int size = 5;
  int marks1[5] = {90, 80, 20, 30 , 60};
  int marks2[5] = {40, 50, 60, 70 , 80};
  Student s1("Ali", 20,2,1999, marks1);
  Student s2("Hamza", 5,1,1998, marks2);
  s1.Print();
  s2.Print();
}
```

Output/Error:

Ali 5-1-1998
40 50 60 70 80
Hamza 5-1-1998
40 50 60 70 80
Error:

1- Line: `delete[] marks; // Static data being deleted multiple times`

2- Line: marks = new int[5]; // Memory Leakage

Part(ii)

```
class B{
private:
	int* bptr;
public:
	B(int b=10){bptr = new int(b);}
	virtual int GetValue(){
		return *bptr;
	}
	virtual ~B(){
		cout<<"~B()    ";
		if(bptr != 0) delete bptr;
	}
};
class D1 : public B{
private:
	int* dptr1;
public:
	D1(int d1=20){dptr1 = new int(d1);}
	int GetValue(){
	   return (B::GetValue() + *dptr1);
	}
	void Print(){
	   cout<<"*dptr1 = "<<*dptr1<<endl;
	}
	~D1(){
		cout<<"~D1()    ";
		if(dptr1 != 0) delete dptr1;
	}
};
class D2 : public B {
private:
	int* dptr2;
public:
	D2(int d2=30){
		dptr2 = new int(d2);
	}
	int GetValue(){
	   return (B::GetValue() + *dptr2);
	}
	~D2(){
		cout<<"~D2()    ";
		if(dptr2 != 0) delete dptr2;
	}
};
class GC : public D1{
private:
	int* gcPtr;
public:
	GC(int gc=40) : D1(gc+10){
		gcPtr = new int(gc);
	}
	int GetValue(){
	   return (D1::GetValue() + *gcPtr);
	}
	void Print(){
	   cout<<"*gcptr = "<<*gcPtr<<endl;
	}
	~GC(){
		cout<<"~GC()    ";
		if(gcPtr != 0) delete gcPtr;
	}
};
void main()
{
	B* arr[4];
	arr[0] = new B(1);
	arr[1] = new D1(2);
	arr[2] = new D2(3);
	arr[3] = new GC(4);

	for(int i=0; i<4 ; i++)
	{
	   cout<<arr[i]->GetValue()<<" , ";

	}
	cout<<endl;
	for(int i = 0 ; i<4 ; i++)
	{
		delete arr[i];
		cout<<endl;
	}
	cout<<"-------------------------\n";
	D1* arr2[2];
	arr2[0] = new D1(100);
	arr2[1] = new GC(500);
	for(int i = 0 ; i<2 ; i++)
		arr2[i]->Print();
	for(int i = 0 ; i<2 ; i++)
	{
		delete arr2[i];
		cout<<endl;
	}
}
```

Output/Error:

```
1, 12, 13, 28
~B()
~D1() ~B()
~D2() ~B()
~GC() ~D1() `B()
---------------------------
```

```
*dptr1 = 100
*dptr1 = 510
~D1() ~B()
~GC() ~D1() ~B()
```

Part(iii)

```
void RecFun(int* arr, int start, int end)
{
	if(end > start+1)
	{
		int mid = (start+end)/2; //int(4.5) = 4
		cout<<"mid = "<<mid<<"\t arr[mid] = "<<arr[mid]<<endl;
		if(arr[mid]%2 == 0)
		{
			RecFun(arr, start, mid-1);
			cout<<"Value = "<<arr[mid]<<endl;
			return;
		}
		else
		{
			RecFun(arr, mid+1, end);
		}
		for(int i=start; i<mid; i++)
			cout<<arr[i]<<",";
		cout<<endl;
	}
}
void main()
{
	int arr1[] = {1,2,6,9,5,7,12,8,9,10};
	RecFun(arr1, 0, 9);
}
```

Output/Error:

```
Mid = 4		arr[mid] = 5
Mid = 7		arr[mid] = 8
Value = 8
1, 2, 6, 9
```