WEEKLY ASSIGNMENT 4

Total Time: 3 Hrs

Total Marks: 100

I. Short Questions (Best of 10) (Total Marks: 10x2 = 20)

1. Explain with example what is a thread?
2. Explain with example what is the difference between asynchronous programming and parallel programming?
3. What is the difference between an action and a func delegate explain with example?
4. What is the difference between normal throw and throwing the exception object?
5. Write a function and use a predicate object to call it?
6. Explain with example the difference between a Thread.Start() and Task.Run() methods?
7. What are the different methods of ensuring safety of cross thread data access, explain with example?
8. How can you make a thread run synchronously explain with example?
9. What is the difference between a thread and a process explain with example?
10. How do you pass a parameter to a thread explain with example?
11. What is the difference between Thread.Suspend() and Thread.Abort() explain with example?

II. Objective Questions (15x1= 15 Marks)

1. C# supports which kind of Multitasking:

a) process based

b) thread based

c) only process

[✓ d)] both process & thread based

2. Choose the correct statement about process-based multitasking.

a) A feature that allows our computer to run two or more programs concurrently

b) A program that acts as a small unit of code that can be dispatched by the scheduler

c) Only A program that acts as a small unit of code that can be dispatched by the scheduler

[✓ d)] Both A feature that allows our computer to run two or more programs concurrently & A program that acts as a small unit of code that can be dispatched by the scheduler

3. Choose the statements which indicate the differences between the thread-based multitasking and process-based multitasking.
[✓ a)] Process-based multitasking handles the concurrent execution of programs
b) Process-based multitasking handles the concurrent execution of pieces of the same program
c) Thread-based multitasking handles the concurrent execution of programs
d) Thread-based multitasking deals with the concurrent execution of pieces of the same program

4. Number of threads that exists for each of the processes that occurs in the program:
a) at most 1
[✓ b)] at least 1
c) only 1
d) both at most 1 & at least 1

5. Which of these methods of Thread class is used to Suspend a thread for a period?
[✓ a)] sleep()
b) terminate()
c) suspend()
d) stop ()

6. What is the advantage of the multithreading program?
a) Enables to utilize the idle and executing time present in most programs
b) Enables to utilize the idle time present in most programs
[✓ c)] Both Enables to utilize the idle and executing time present in most programs & Enables to utilize the idle time present in most programs
d) Only Enables to utilize the idle time present in most programs

7. What does the following method specify?
Type[] GetGenericArguments()
a) A property defined by MemberInfo
[✓ b)] Obtains a list of the type arguments bound to a closed constructed generic type
c) The list may contain both type arguments and type parameters
d) All of the mentioned

8. Which of these statements is incorrect?

a) By multithreading CPU's idle time is minimized, and we can take maximum use of it

b) By multitasking CPU's idle time is minimized, and we can take maximum use of it

c) Two threads in C# can have same priority

[✓ d)] A thread can exist only in two states, running and blocked

9. What is multithreaded programming?

a) It's a process in which two different processes run simultaneously

[✓ b)] It's a process in which two or more parts of same process run simultaneously

c) It's a process in which many different processes are able to access same information

d) It's a process in which a single process can access information from many sources

10. Choose the advantages of using generics?

a) Generics facilitate type safety

b) Generics facilitate improved performance and reduced code

c) Generics promote the usage of parameterized types

[✓ d)] All of the mentioned

11. What will be the output of the following C# code snippet?

```
public class Generic<T>
{
    Stack<T> stk = new Stack<T>();
    public void push(T obj)
    {
        stk.Push(obj);
    }
    public T pop()
    {
                T obj = stk.Pop();
                return obj;
            }
        }
        class Program
        {
            static void Main(string[] args)
            {
                Generic<string> g = new Generic<string>();
                g.push(40);
                Console.WriteLine(g.pop());
                Console.ReadLine();
            }
        }
```

a) 0

[✓ b)] Runtime Error

c) 40

d) Compile time Error

12. What will be the output of the following C# code snippet?

```
public class Generic<T>
{
    Stack<T> stk = new Stack<T>();
    public void push(T obj)
    {
        stk.Push(obj);
    }
    public T pop()
    {
                T obj = stk.Pop();
                return obj;
            }
        }
        class Program
        {
            static void Main(string[] args)
            {
                Generic<int> g = new Generic<int>();
                g.push("Csharp");
                Console.WriteLine(g.pop());
                Console.ReadLine();
            }
        }
```

[✓ a)] Compile time error

b) Csharp

c) 0

d) Run time error

13. What is meant by the term generics?

[✓ a)] parameterized types

b) class

c) structure

d) interface

14. Choose the correct way to call subroutine fun() of the sample class?

```
class a
{
    public void x(int p, double k)
    {
        Console.WriteLine("k : csharp!");
    }
}
```

a)

```
delegate void del(int i);
```

```
x s = new x();
del d = new del(ref s.x);
d(8, 2.2f);
```

[✓ b)]

```
delegate void del(int p,  double k);
del d;
x s = new x();
d = new del(ref s.x);
d(8, 2.2f);
```

c)

```
x s = new x();
delegate void d = new del(ref x);
d(8, 2.2f);
```

d) all the mentioned

15. What will be the output of the following C# code?

```
public class Generic<T>
{
    public T Field;
}
class Program
{
    static void Main(string[] args)
    {
        Generic<int> g2 = new Generic<int>();
                Generic<int> g3 = new Generic<int>();
                g2.Field = 8;
                g3.Field = 4;
                if (g2.Field % g3.Field == 0)
                {
                    Console.WriteLine("A");
                }
                else
                Console.WriteLine("Prints nothing:");
                Console.ReadLine();
            }
        }
```

a) Compile time error

[✓ b)] A

c) Run time error

d) Code runs successfully but prints nothing

III. Programming Questions (Answer any 4) – (15x4 = 60)

1. Write a program which facilitates the following

- The program should give an option to the user a few options.

- The user can choose to create a thread. Make sure the threads are infinitely running threads
- The user can choose to destroy a thread by selecting a thread ID.
- The user can decide to see how many threads are running.
- The user can decide to either make a thread run synchronously by ID
- The user can decide to make a thread sleep for a specific number of seconds.

2. Write a C# program which can do the following operations.
   - The program facilitates a few operations which can be actuated by the specific classes.
   - The program facilitates an add method which can be called with two integer parameters, two string parameters, two List parameters and two custom class parameters. For two integers it returns the sum, for two strings it concatenates, for two lists it joins the Lists and for two custom classes it again merges their property values using a list and returns.

3. Write a C# program which has the following construct
   - The program has an Interface which implements three generics type. And It has two Methods Validate and Inspect.
   - The Validate Method takes first generic type as Input and returns a Boolean value.
   - The Inspect method takes both the generic type as Input and returns the third generic Type as output.
   - Create a generic class which implements the interface. You can just print the Type of arguments passed in those methods.
   - Now Implement Create object of the class and call the class with different type of parameters.

4. Write a Program which implements a calculator functionality
   - A calculator has Add, Divide, Multiply, Subtract, Remainder, Power operations.
   - All these operations from taking user input to actual operations work on a separate thread.
   - All inputs are taken from the user and corresponding exceptions are handled. The exceptions are thrown from the corresponding threads to the main calling threads.
   - However, the calling thread only catch a single type of exception and not all different exceptions. It displays the relevant message passed and shows the complete stack where the error occurred.

5. Write a C# program which creates a Generic List like implementation called MyList<T>
   - It must expose Add(T). Remove(T), RemoveAt(int i), RemoveAll(), IndexOf(T) methods
   - You cannot use the List<T> and create a wrapper, implement the functionality from scratch.
   - You should also throw any relevant exceptions to the calling class.

IV. Design Question (1x5 = 5 marks)

1. Design a data storage system:
   - The system facilitates storing of data from multiple users at the same time.
   - The users need authenticate to access and modify any data.
   - The data once sent to the server need to be queued for processing and stored.
   - The data storage must facilitates roll back mechanism in case of failure.
   - The data must be stored in such a way that it facilitates an optimal storage and retrieval time.

   Use the concepts that were taught in the class and design a complete shopping module. No coding expected only design.