|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Event Driven Programming** | | | | |
| **Year** | **No.** | **Question** | **Marks** | **Answers** |
| 18/19  (Main) | 4 | Within the context of event-driven programming, the job of the “Dispatcher” is to take each event that comes to it, analyse and process it further. Usually these systems do have an ‘Event Queue’. Why? [10 marks] | 10 | Event queue is used to archive events when dispatchers or managers cannot handle events as quickly as they arrive. The event queue is brought into the event stream where the stream can be temporarily kept. When an event arrives, it is put into the end of the queue and the dispatcher handles events in the front of the queue as quickly as possible. Most GUI Applications typically include an event queue. Larger events like mouse clicks can takes longer to be handled. While this happens, other events such as mouse movement events can accumulate in the buffer. When the dispatcher becomes free again, it can speedily discard the ignorable mouse movement events and quickly clear the event queue. |
| 18/19 | 4 | Within the context of event-driven programming, the job of the “Dispatcher” is to take each event that comes to it, analyse and process it further. Why do such systems usually have an ‘Event Queue’? [10 marks] | 10 |
| 17/18  (Main) | 4 | 1. Explain why embedded systems are a typical application area for event driven programming. [10 marks] 2. Describe the role of the ‘Dispatcher’ within the context of event driven programming. [10 marks] | 20 | <https://drive.google.com/drive/folders/1Dof0qKgvjpdZ_HBfsDVZW-6VTBzlFrEz> (7-8 Pages) |
| 17/18 | 4 | What is the role of the ‘Dispatcher’ within the context of event driven programming? [10 marks] | 10 | The job of the dispatcher is to take each event that comes to it, analyse the event to determine its event type, and then send each event to a handler that can handle events of that type. |
| 16/17 | 4 | GUI’s (Graphical User Interfaces) are a typical application area for event driven programming. Why? [9 marks] | 9 | * **Best for Graphical User Interface:** Event driven programming in best technique for GUI applications, because in GUI applications user interacts with the system using events (Mouse Click, Key Press etc.). * [Event driven programming](https://www.locusassignments.com/units/unit-20-event-driven-programming-solutions) can also be used with procedural languages where all event handling can be defined as separate procedures. * **Service Oriented:** In event driven programming many of our event handlers are service oriented. Service oriented event handlers works as background services and serve users silently in the background. * **Time Driven:** In **event driven programming** we can define time driven event handler. Time driven event handler execute automatically after a specific time interval.   Features of Event Driven Programming:   * **Flexibility:** Event driven programming is very flexible and a wide range of applications can be created using event driven programming. * **Suitable for graphical interfaces:** Event driven programming is very suitable for graphical interfaces because in GUI application user interact with the system via events. * **Simplicity of programming:** Event driven programming is very simple compare to traditional flow driven languages. * **Ease of development:** Event driven programming provide easy development environment for programmer because he/she only need to concentrate only on one event a time   <https://www.locusassignments.com/solution/unit-20-event-driven-programming-solutions-assignment> |
| 16/17 | 4 | Embedded systems are a typical application area for event driven programming. Why? | 9 | **Event driven programming** is very suitable for **embedded systems** because many **embedded systems** are **event driven** by nature. Most of the tiny embedded systems respond to external or internal events in some or other way. The external event can be an interrupt, or change of signal level at I/O pins.  Milans Answers>>  These days millions of PCs and embedded devices are not batch processing, almost all computers are event-driven. Thus, event-driven programming is most suitable for embedded systems because the embedded systems are event driven by nature. The Event can be a button press, sensor detection, etc. As an example, think of a printer or elevator.  file:///C:/Users/ACER/Downloads/icons\_2011\_1\_40\_20083.pdf |
| 14/15 | 1 | 1. Describe a typical application area where event driven programming is useful and relevant. [8 marks] 2. Assume you are a Java developer. On your application you notice that pressing a button in the GUI of your application generates an error. Using your debugging tools you notice that the error seems to originate from a class “EventDispatchThread” in the “java.awt” package (see Figure 1): 3. Discuss the purpose and context of this class “EventDispatchThread” [10 marks] 4. Formulate an opinion where to look for the possible error in the code. [7 marks] | 25 | Event-driven programming is applied extensively in graphical user interfaces and applications that perform certain actions in response to user input (for example, JavaScript web applications).  <https://www.bbconsult.co.uk/blog/event-driven-applications>  A *thread* is a single sequential flow of control within a program. A single thread also has a beginning, a sequence, and an end. At any given time during the runtime of the thread, there is a single point of execution. However, a thread itself is not a program; a thread cannot run on its own. Rather, it runs within a program.  EventDispatchThread is a package-private AWT class which takes events off the EventQueue and dispatches them to the appropriate AWT components. The Thread starts a "permanent" event pump with a call to pumpEvents(Conditional) in its run() method. Event handlers can choose to block this event pump at any time, but should start a new pump (not a new EventDispatchThread) by again calling pumpEvents(Conditional). This secondary event pump will exit automatically as soon as the Condtional evaluate()s to false and an additional Event is pumped and dispatched.  <https://www.iitk.ac.in/esc101/05Aug/tutorial/essential/threads/definition.html> |
| 11/12 | 4 | In event driven programming which software pattern is typically used to implement a subscribe/notify behaviour of the system where handlers subscribe to a subject that notifies the handlers of events of interest to them? | 2 | Observer pattern  <https://en.wikipedia.org/wiki/Observer_pattern> |
| 10/11 |  | In event driven programming which software pattern is typically used to implement a subscribe / notify behaviour of the system where handlers subscribe to a subject that notifies the handlers of events of interest to them? | 2 |

